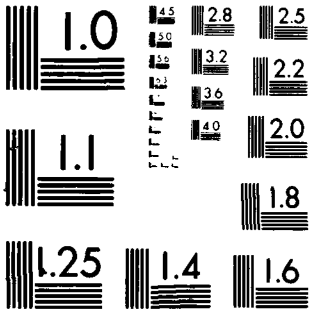
AD-A111 753   PENNSYLVANIA TRANSPORTATION AND TRAFFIC SAFETY CENTER--ETC   F/G 13/10
GREAT LAKES SIMULATION STUDIES. VOLUME I. NETSIM: A GENERAL NET--ETC(U)
NOV 72   DACW23-72-C-0066
UNCLASSIFIED   TTSC-7215   NL

1 of 2
[illegible]



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

1

THE PENNSYLVANIA STATE UNIVERSITY

GREAT LAKES SIMULATION STUDIES

VOLUME 1

NETSIM: A GENERAL NETWORK SIMULATOR

INTERIM REPORT

CONTRACT NO. DACW23-72-C-0066

NOVEMBER 1972

IN COOPERATION WITH THE DEPARTMENTS OF BUSINESS LOGISTICS,
CIVIL ENGINEERING, AND MANAGEMENT

PENNSYLVANIA TRANSPORTATION
AND TRAFFIC SAFETY CENTER

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Great Lakes Simulation Studies. V.1 NETSIM: A General Network Simulator.		5. TYPE OF REPORT & PERIOD COVERED Interim Report 1972
		6. PERFORMING ORG. REPORT NUMBER TTSC No.7215
7. AUTHOR(s)		8. CONTRACT OR GRANT NUMBER(s) DACW23 72C 0066
9. PERFORMING ORGANIZATION NAME AND ADDRESS Pennsylvania. State University. Transportation and Traffic Safety Center.		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Engineer Division, North Central Planning Division. Economics Branch. 536 S. Clark St./Chicago, IL 60605		12. REPORT DATE November, 1972
		13. NUMBER OF PAGES 105 p.
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release. Distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Approved for public release. Distribution unlimited.

18. SUPPLEMENTARY NOTES

Library of Congress Number: TA1145 P412ttsc No.7215 Assigned by controlling office.

Series title: Great Lakes Simulation Studies. Series title varies.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

1. TRANSPORTATION MODELS 2. COMPUTERIZED SIMULATION 3. GREAT LAKES 4. ST. LAWRENCE SEAWAY 5. NETSIM/SHIP 6. *MULTIPLE CHANNEL NAVIGATION 7. WATERWAY TRANSPORTATION

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

In developing a computer model for the simulation of the Great Lakes and St. Lawrence Seaway navigation system, this report concentrates on a thorough explanation of NETSIM (a general network simulator) with its input-output description, a good section on model structure and the NETSIM/SHIP user's manual. The word 'SHIP' was added due to NETSIM's use with multiple channel deep draft navitation systems.

ABSTRACT

NETSIM is a computer simulation model written in the SIMSCRIPT programming language whose purpose is to provide a general simulation capability for any network composed of links and nodes. The current use of NETSIM in multiple channel deep draft navigation systems has earned it the qualification NETSIM/SHIP. The model was developed for use by the U.S. Army Corps of Engineers with immediate applications on the Great Lakes systems.

Ships are handled in NETSIM/SHIP on a chronological event basis as they engage in their journey from origin to destination. Simulation of ship processing at system facilities (locks, reaches, etc.) is carried out through the use of distinct link modules, and channel choice decisions where alternative routes exist are handled by a decision-making mechanism based on the calculation of expected transit times for each route.

Required inputs consist of fleet data, run and system size parameters, and a description of the network configuration and system entities. Output produced by the model consists of data for the formulation of the channel choice mechanism in the first form, and an event by event description of the actual simulation in the second form. The latter form provides a permanent data base from which exactly tailored statistical reports can be generated.

Although the model is complete as it currently exists, it is being further equipped with extensive capabilities for general, system wide simulation of the Great Lakes. Even without these capabilities, however, the current version of NETSIM/SHIP can be used as a planning tool to aid in the investigation of subsystem operating characteristics.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

The work described in this interim report was performed by the Pennsylvania Transportation and Traffic Safety Center (PTTSC) at The Pennsylvania State University for the U.S. Army Corps of Engineers, North Central Division, under contract number DACW-23-72-C-0066. The contract period is from 1 July 1972 to 31 June 1973.

The effort reported herein represents a continuation of work at PTTSC in the general area of waterway systems planning and analysis. Earlier research was devoted to the development of a simulation model for applications on the inland waterways, and to the use of the model to study structural and nonstructural improvements. This report, in documenting the provision of deep draft navigation systems simulation capability, extends the horizon for complex system research and planning.

Dr. Joseph L. Carroll served as the Principal Investigator for this task. Dr. John C. Rea, David C. Nowading, and P. Wade Buckholts provided the conceptualization and framework for the model. Robert A. MacLaughlin, Economics Branch, North Central Division provided technical liason and guidance for the Corps.

A complete list of contributors appears below.

Joseph L. Carroll
Head, Transportation Systems Division, PTTSC
Professor of Business Administration

Michael S. Bronzini
Research Assistant, Department of Civil Engineering and PTTSC

Srikanth Rao
Research Assistant, PTTSC

P. Wade Buckholts
Graduate Assistant, PTTSC

John C. Rea
Head, Urban and Environmental Research, PTTSC
Assistant Professor of Civil Engineering

David C. Nowading
Research Assistant, PTTSC

Special acknowledgement is made of the services and guidance provided by the following people and organizations:

(1) Mr. M. S. Campbell, Mr. Kosky, and Mr. Ahermy of the St. Lawrence Seaway Authority, St. Catherines, Ontario, Canada.

(2) Mr. R. Goodno and Mr. H. Shaudy, U.S. Army Corps of Engineers, Buffalo District.

(3) Dr. Paul F. Wyman, Assistant Professor of Management Science, The Pennsylvania State University.

I. INTRODUCTION

In June of 1972, the Army Corps of Engineers, North Central Division entered into a contract with the Pennsylvania Transportation and Traffic Safety Center for the development of a simulation model that would facilitate a systematic analysis of the capacity of the Great Lakes Navigation System. The development of this simulation model was to undergo three phases:

1. To develop a LE-LO Navigation Simulation model;
2. To apply the model to simulation studies of the Welland Canal and proposed alternatives to the Welland;
3. To revise the initial simulation model so as to include the capabilities needed for comprehensive Great Lakes-St. Lawrence System Simulations.

This report presents a description of the first phase, i.e. the LE-LO Navigation Simulation model. The description encompasses the underlying conceptual structure of the model as well as the logical flow of the program. A user's manual is presented in the Appendix to describe in detail the data needs and preparation.

A. HISTORICAL BACKGROUND

The construction of the LE-LO Navigation Simulation Model dates back to the use of computerized simulation models on the inland waterways. The circumstances leading to their development and the results of that research are reported in a six-volume technical report entitled Waterway Systems Simulation (1). The groundwork for Greak Lakes simulation was laid in Volume V of this series, entitled Simulation of Multiple Channel Deep Draft Navigation Systems (2). The model presented in Volume V will henceforth be referred to as the MCDD model (Multiple Channel Deep Draft).

The development ot the MCDD model was commissioned by the Corps of Engineers to provide simulation analyses of the Great Lakes-St. Lawrence Seaway transportation system. The study had two objectives: first, to build a simulation model specifically designed to simulate the performance of a multiple channel deep draft ship canal in which each channel consists of a series of locks and reaches; and, second, to formulate a methodology for assigning vessels between parallel canals.

The conceptual basis for the MCDD model's multiple, parallel channel simulation capability is its assignment decision technique. The methodology employed by the model for determining a vessel's expected transit time through a canal is not dissimilar to the decision process that would be undertaken by a rational and experienced channel controller (i.e. one who assigns each vessel to a channel). When a vessel approaches the channel choice point the controller, given the current channel conditions, could draw upon his experience to predict the vessel's probable transit time through each channel The vessel would then be assigned to the channel offering the least expected transit time.

The form of the decision-making mechanism whether it be the channel controller, the vessel captain or pilot, or simply some inanimate information system is actually irrelevant as regards to the justification for the methodology employed by NETSIM/SHIP. The critical assumption is that a state-dependent relationship for the expected transit time be obtainable.

The MCDD model simulates the channel controller's experience by using a priori relationships between prior conditions and subsequent transit times. These relationships exist in the form of decision functions statistically derived from an experience data base. To develop an Experience Data Base (EDB), a user arbitrarily selects one canal within a set of parallels through which all transits are to move During a simulation, the EDB mechanisms produce, as output, the conditions existing within each arbitrarily selected canal when a vessel approaches the assignment decision point. Succinctly, the EDB contains canal conditions prior to a vessel's transit. As each vessel exits a canal, the EDB mechanisms again produce output--this time in the form of the time spent by a vessel in transiting a canal, i.e., a vessel's subsequent transit time. The EDB, containing prior canal conditions and subsequent transit times, is statistically analyzed to derive functions for predicting expected transit times. A function is derived for each parallel canal within a simulated system. The set of functions is then included with the model in a following simulation, where it is used to predict expected transit times.

Thus, using statistically derived expected transit time functions, a vessel is assigned to a channel based upon least expected transit time. Once a vessel is assigned to a channel, it receives an assignment map which indicates the sequence of reaches and locks to be followed. Using the vessel's map, a movement control mechanism monitors the vessel's course while in transit through the series of reaches and locks.

The lock simulator included in the MCDD model simulates actual lock operations by sampling, at appropriate times, from seven probability distributions which describe the different time elements in the lock's operations. The locks use an alternating service queue discipline. The reach simulator samples from two probability distributions, thereby considering any possible effects that direction of transit might have upon reach transit time. Vessels are not allowed to pass other vessels during reach transits

The MCDD model, as described by its authors is intended for simulating alternative channel configurations for comparative analysis. The scope of the model does not include the capability for total system studies which, for some time (3) has been considered essential, as, indeed, it still is (4) if alternative waterway improvements are to be meaningfully evaluated. The authors suggest modifications that, if implemented, would develop the MCDD model into a general systems model capable of simulating, for example, the Great Lakes-St Lawrence transportation system. Suggestions are also given for refining the lock simulator so that the model could be used for lock design studies.

The authors suggest the addition of three features, listed as follows, which would transform their model into an effective lock design tool:

1. The ability to differentiate by vessel size or weight and select appropriate distributions.
2. The ability to differentiate by movement direction.
3. The ability to "look ahead" to determine the next vessels to be serviced according to some operating criterion.

These three additional features serve to increase the complexity of the model, as its realism is increased. For increasing the model's generality, two additional features, mentioned below, are suggested:

4. The capability to initiate and terminate vessel movements within reaches.
5. The capability to simulate the time spent in port by a vessel as a function of port conditions, facilities, cargo, etc.

The ability to differentiate vessels by their primary characteristics allows research studies to be conducted covering the relationships between these vessel characteristics and system utilization. Another important consequence of this feature is the ability to study the effects of vessel characteristics (especially size and speed) in relation to lock design. This relationship becomes increasingly important as vessels increase in size and speed, since locks have an expected life of around fifty years (5).

Because of the possible effects of water flow direction upon the speed of vessels, it is desirable to be able to sample from distinct probability distributions to simulate those effects. Having separate distributions for each direction also permits a more realistic simulation of the different volumes of vessel transits moving upstream and downstream.

To use the model for lock design studies, the lock simulator must closely simulate actual lock operations. Inclusion of a "service lookahead" feature provides the ability to simulate the actions that an intelligent experienced lockmaster would take. These actions pertain particularly to the situation when the lock is idle, or empty. The lockmaster scans the adjacent reaches and, when he sees an imminent arrival, commits the lock, recycling it necessary, in order to minimize the arrival's delay. When arriving vessels exist in both reaches the experienced lockmaster determines which vessel will probably arrive first, taking into consideration the time required for recycling, and commits the lock to that vessel expected to arrive first.

To be able to simulate, for example, a shipping season in the Great Lakes-St. Lawrence Seaway transportation system, the model must be capable of beginning its vessels' transits from different ports at different times. Simulation of the closing of the shipping season requires that vessels be able to terminate their transits at any ports in the system. This feature also has the side benefit of allowing a simulation to begin in near steady state, given that an acceptable definition of steady state is available.

The model obviously must include simulators of all the facilities comprising a given system To permit the simulation of the Great Lakes-St. Lawrence Seaway, for example, the model must include port and lake simulators, as well as its current lock and reach simulators. The port simulator should have, as a minimum, the ability to accept a vessel and hold it for some random time period based upon attributes of the vessel and of the port The lake simulator should be able to simulate an actual lake situation, with multiple ports and entry/exit channels.

B. THE LE-LO NAVIGATION SIMULATION MODEL

After reviewing the above suggestions, the COE authorized the PSU analysis group to undertake a development effort whereby these suggestions would be implemented The resulting model is designated by the acronym NETSIM (NETwork SIMulator) and due to its current applications is further qualified as NETSIM/SHIP

NETSIM/SHIP has been programmed with a particular eye towards flexibility. This flexibility is especially due to four factors. First, flexibility is rendered by the modular design which allows the insertion, removal or extension of any program segments of the model to make it adaptable for a given set of extenuating circumstances. To emphasize this modular design a brief explanation is provided.

NETSIM/SHIP is composed of four separate logic modules. The first two modules are concerned with system initialization and traffic control functions respectively The third module consists of four distinct sub-modules, one for each of NETSIM/SHIP's four facilities (lock, lake, reach, and port). The fourth logic module provides utility programs in support of the other three modulse Thus the modification of any of these modules (or submodules) to further tailor it for particular circumstances is greatly facilitated.

Flexibility is further attained by virtue of the fact that the model is so implemented on the Penn State computer facility such as to allow remote operation of the model. That is, operation is not confined to the Penn State campus, but rather is limited only by the location of a suitable communications terminal and a voice-grade telephone line.

Flexibility also exists in the use of the computer programming language used to encode the model. The MCDD model is written in GPSS, while NETSIM/SHIP is written in SIMSCRIPT. GPSS is a problem-oriented language written specifically for users with little or no programming experience and its simplified structure results in some loss of flexibility. On the other hand, SIMSCRIPT requires more programming skill, but it is capable of representing more complex data structures and can execute more complex decision rules. Further, its English-like readability allows a SIMSCRIPT written model to serve to a great extent as its own documentation. Thus the flexibility of SIMSCRIPT can be summarized by saying that in complex models, SIMSCRIPT is able to produce a more compact model that requires less storage space, and that generally will be executed more rapidly.

Flexibility is also rendered by the use of a three-phase approach in executing the model. These phases may properly be termed as input, execution, and output analysis. The input phase is discussed first.

A rather extensive preparation of the input data stream becomes a prerequisite to using the model because of the technical complexity of the subject areas (facility design, commodity transfers, ship scheduling, etc.) of potential NETSIM SHIP applications, and because no ubiquitous data base is available. The heart of this input stream is the fleet data. NETSIM/SHIP provides for the input of these fleet data in either an endogenous or an exogenous manner. The latter is of more crucial interest, since it allows the use of the output from such a complex vessel generation program as TOWGEN (6). The input data stream consisting of the fleet data serves as the first phase.

The actual simulation using NETSIM/SHIP constitutes the second phase. This phase itself is composed of two sub-phases: (1) the EDB phase and (2) the Event LOG phase. The sub-phase structure of the second (simulation) phase is necessitated by NETSIM/SHIP's alternative-selection functions. When, in transiting a simulated network, a vessel approaches a point where alternative routes exist, NETSIM/SHIP's dynamic route determination logic selects one of the available alternatives on the basis of least expected transit time. The expected transit time for each available alternative is calculated by feeding data describing current traffic conditions to a set of expected transit time functions. These expected transit time functions contain coefficients which weight the various traffic conditions. It is these coefficients that are derived from the first, or EDB, sub-phase of NETSIM/SHIP's simulation phase. The EDB phase determines the coefficients for the expected transit time functions by simulating a network configuration with selected alternative routes removed from the system. In this way, all transits are forced to use a particular alternative route. Regression analysis is then used to determine the values of each coefficient in the function for a specific selected alternative.

Once all the expected transit time functions needed to simulate a particular network have been derived, the second, or EVENT LOG, sub-phase can be conducted. The expected transit time functions are utilized during the EVENT LOG phase to dynamically consider the effects of current traffic conditions upon expected transit time. The EVENT-LOG sub-phase generates a log of each event occurrence during simulation. It is this event log which is used during the third phase of the modeling.

Because of the complexity and sheer volume of NETSIM/SHIP's encoded logic structures, it was decided to remove the burden of statistical evaluation to a post-simulation phase. Thus the third phase consists of a post-simulation processor program which utilizes the event log generated during the second phase to summarize significant statistical data.

It may be noted that the generation of an event log provides a secondary advantage in the creation of a permanent, detailed record of the simulation. The post processor can therefore be tailored to fit the needs of the user and may simply be modified if more information is desired. The latter step does not require a run of the time-consuming simulation phase.

II. INPUT-OUTPUT DESCRIPTION

This chapter is designed to provide a quick summary of NETSIM's data requirements and explain some of the more complicated sections. An item by item description of the input and output data sets is also given in the NETSIM/SHIP USER'S MANUAL provided in the Appendix.

A. INPUT DATA BASE

NETSIM/SHIP's input data stream is broken into the following four categories:

1. Run parameters
2. System size parameters
3. System entity descriptors
4. Network configuration descriptors

1. Run parameters

The first category of input data, the run parameters, provides NETSIM/SHIP with information pertinent to certain program options, input and output devices, and the network configuration to be simulated. These include the following

a. Shipping season length to be simulated

b. Simulation run type (whether EDB or EVENT LOG) specification

c. Service look ahead feature specification

d. Simulation of parallel alternatives specification

e. Output unit device

f. Input unit device for exogenous data

g. Destination of vessel upon port entry specification

h. External data specification

2. System size parameters

The size parameters include seven data elements that together describe the extent of the network being simulated. These parameters consist of the number of ports, lakes, reaches, locks, vessels, nodes and the number of vessels with predetermined itineraries.

3. System entity descriptors

NETSIM/SHIP includes five different system entities: (a) ports, (b) lakes, (c) reaches, (d) locks, and (e) vessels. A description of the data requirements for each entity follows.

Each port within a network requires five data elements. The first two elements consist of the port identification, and a commodity code representing the ability of the port to process the commodity. The third element is a frequency distribution giving the probabilities of a vessel's movement from the current port to any other port in the network. This element was included in the model to provide a basis for future experimentation of a vessel schedule based on random access of ports. The fourth element in the port descriptors specifies whether the frequency distribution for port turnaround time is empirically supplied or is to be derived from a SIMSCRIPT supplied theoretical probability distribution. The fifth element, therefore, consists either of an empirical frequency distribution or the arguments for a theoretical distribution.

A lake requires, minimally, six data elements. The first two elements comprise the lake identification and the number of nodes, ports included, associated with the lake. The third element specifies whether the lake is one branch of a set of parallel routes. The fourth element identifies the lake transit time distribution as being either theoretical or empirical.

The fifth element is the distribution itself if empirically given or the arguments if theoretically specified. The sixth element is required only if the simulation run is of an EVENT LOG type and if the lake is in a parallel set. In such a case, the sixth element specifies the effects of the lake conditions in the calculation of the expected transit time for a vessel through that alternative. The seventh element consists of an origin-destination (O-D) table providing the distance between any two nodes (including ports) on the lake.

Each reach in the system configuration requires at least nine elements. The first two elements are the reach identification number and the length of the reach. The third specifies whether passing is allowed in the reach. The fourth specifies whether the reach is part of a parallel set. The fifth element indicates whether the reach transit time distribution is theoretical or empirical. The sixth and the seventh elements constitute the end point nodes for the reach and the eighth and ninth elements provide the reach transit time distributions, one for each direction of travel. If the simulation is of an EVENT LOG type and if the reach is in a parallel set, then two more elements are required, one for each direction. These specify the effects of the reach conditions in the calculation of the expected transit time for a vessel through that alternative.

The first element of the set of lock descriptors is its identification number. The second and third elements provide the two end point nodes of the lock while the fourth element specifies the maximum vessel length permitted in the lock. The fifth element indicates whether the lock is in a parallel set of alternative routes. The next nine elements are associated with nine time blocks constituting the lock frequency distributions. The sixth through fourteenth elements indicate whether their associated time blocks (i.e. distributions) are supplied empirically or theoretically. The next nine elements, therefore, provide the actual time blocks, either empirical distributions or theoretical arguments.[1] Two more elements are supplied if the simulation is of an EVENT LOG type and if the lock is part of a parallel set. These describe the effects of the lock conditions in the calculation of the expected transit time for a vessel through that alternative in each direction.

Vessel data can be supplied either through the regular input data stream or from an external input device during simulation. The vessel data consists of the vessel identification number, its length, a commodity code representing the vessel's load, its origin port, an itinerary code and its time of departure from the origin port. Further, if the vessel has a pre-defined itinerary, this is also specified. The exact order of these parameters depends on whether the vessel data are externally supplied or not.

4. Network configuration descriptors

There are three data tables required by NETSIM/SHIP that together comprse a map of the network configuration being simulated. The first, the TABLE OF NEXT NODES, is essentially an Origin-Destination (O-D) matrix which also provides information regarding the availability of alternative routes. The FACILITIES ID TABLE, the second required data table, provides the identification number for each facility within a network configuration. Whenever a configuration contains alternative parallel routes, these alternatives are described in the third table, the PARALLEL FACILITIES TABLE.

[1] NETSIM/SHIP can currently differentiate the locking times between three vessel classes.

If the configuration consists of alternative parallel routes, a fourth table, ALTERNATIVE WIDE COEFFICIENTS, is also required. This table describes the effects of certain system attributes in the calculation of the expected transit time functions.

The TABLE OF NEXT NODES can be thought of as two separate sections, the first of which is an O-D matrix (refer to Figure 1). The first row of the table contains the identification number of every port within the network configuration being simulated (column 1 of row 1 must be set to 0). This first row identifies every possible destination within a network (vessels' voyages must always terminate at ports). The table's first column contains the identification number for every node (ports as well as non-port nodes) within a network configuration. All network nodes are included in the first column because, during simulation, it is possible for every node to serve as an intermediate origin, from which a vessel seeks a route to its destination. There is no required order in which identification numbers are given, either in the first row or the first column, which means that the upper-left-to-lower-right diagonal does not necessarily contain all zeroes. The body of the table is made of the identification numbers of the next nodes to which vessels must transit in sailing from their current, or origin, nodes (column 1) to their destination ports (row 1). The second section of the TABLE OF NEXT NODES is identical to the first section, but only in its first row and its first column. The body of the second section contains identification numbers, not of next nodes, but of sets of alternative parallel routes. When a network configuration contains alternative parallel routes, they are grouped into sets and each set is assigned a unique identification number. It is these identification numbers that comprise the body of the second section.

Destination Port

	0	PORT ID_1	PORT ID_2	PORT ID_3	...	PORT ID_n
Current Node	NODE ID_1	$NNID^{*}_{11}$	$NNID_{12}$	$NNID_{13}$	...	$NNID_{1n}$
	NODE ID_2	$NNID_{21}$	$NNID_{22}$	$NNID_{23}$	...	$NNID_{2n}$
	NODE ID_3	$NNID_{31}$	$NNID_{32}$	$NNID_{33}$	...	$NNID_{3n}$
	⋮	⋮	⋮	⋮	⋮	⋮
	NODE IDm	$NNID_{m1}$	$NNID_{m2}$	$NNID_{m3}$	...	$NNID_{mn}$

Section One

Destination Port

	0	PORT ID_1	PORT ID_2	PORT ID_3	...	PORT ID_n
Current Node	NODE ID_1	$PSID^{**}_{11}$	$PSID_{12}$	$PSID_{13}$	...	$PSID_{1n}$
	NODE ID_2	$PSID_{21}$	$PSID_{22}$	$PSID_{23}$	...	$PSID_{2n}$
	NODE ID_3	$PSID_{31}$	$PSID_{32}$	$PSID_{33}$	...	$PSID_{3n}$
	⋮	⋮	⋮	⋮	⋮	⋮
	NODE ID_m	$PSID_{m1}$	$PSID_{m2}$	$PSID_{m3}$	...	$PSID_{mn}$

Section Two

*Next Node IDentification Number

**Parallel Set IDentification Number

n = number of ports

m = total number of nodes (ports + non-port nodes)

Figure 1. Table of Next Nodes

If a vessel has a set of alternative routes to choose from in sailing from its current node (the first column) to its destination port (the first row), the identification number of the set of alternatives is entered in the table. If there are no alternatives available to a vessel, a 0 (zero) is entered. For a simulated network configuration which contains no sets of alternative parallel routes, the body of the second section contains all zeroes. The second section nonetheless must be entered with the first section into the input data stream.

The FACILITIES ID TABLE (see Figure 2) contains the identification number of each facility in a network configuration. It, as are the other two data tables discussed in this section, is read into the input data stream by rows. The first row of the FACILITIES ID TABLE contains an identification number for every node in the network, as does the first column. The first row, first column, position is 0 (zero). The body of the table is made up of the identification numbers of the network links through which vessels transit in sailing from a current node (the first column) to a next node (the first row).

Immediately following the FACILITIES ID TABLE, the PARALLEL FACILITIES TABLE (Figure 3) is given if the system consists of parallel routes. Each row of the table may be of a different length, depending upon the number of links comprising the alternative route described by that row. The number of rows in the table is determined by the total number of alternative parallel routes included in the simulated network configuration. Every row in the table has its first four columns in common with the table's other rows. Column 1 contains the identification number of the set of parallel routes. For example, if, between node X and node Y, there are three alternative parallel routes available, these three routes are grouped and

Next Node

Current Node

0	NODE ID$_1$	NODE ID$_2$	NODE ID$_3$	. . .	NODE ID$_n$
NODE ID$_1$	INLID*$_{11}$	INLID$_{12}$	INLID$_{13}$	. . .	INLID$_{1n}$
NODE ID$_2$	INLID$_{21}$	INLID$_{22}$	INLID$_{23}$	. . .	INLID$_{2n}$
NODE ID$_3$	INLID$_{31}$	INLID$_{32}$	INLID$_{33}$		INLID$_{3n}$
⋮	⋮	⋮			⋮
NODE ID$_n$	INLID$_{n1}$	INLID$_{n2}$	INLID$_{n3}$		INLID$_{nn}$

*Inter-Nodal Link IDentification Number

n = total number of nodes (ports + non-port nodes)

Figure 2. FACILITIES ID TABLE

SET ID$_1$	NBR ALT$_1$	EDB ALT$_1$	NBR NODES$_1$	NODE$_{1,1}$	LINK$_{1,1}$	NODE$_{1,2}$	. .
SET ID$_2$	NBR ALT$_2$	EDB ALT$_2$	NBR NODES$_2$	NODE$_{2,1}$	LINK$_{2,1}$	NODE$_{2,2}$	.
SET ID$_3$	NBR ALT$_3$	EDB ALT$_3$	NBR NODES$_3$	NODE$_{3,1}$	LINK$_{3,1}$	NODE$_{3,2}$	
:	:	.					
SET ID$_i$	NBR ALT$_i$	EDB ALT$_i$	NBR NODES$_i$	NODE$_{i,1}$	LINK$_{i,1}$	NODE$_{i,2}$	
			.				.
SET ID$_m$	NBR ALT$_m$	EDB ALT$_m$	NBR NODES$_m$	NODE$_{m,1}$	LINK$_{m,1}$	NODE$_{m,2}$	

	LINK$_{1,j-1}$	NODE$_{1,j}$		LINK$_{1,n-1}$	NODE$_{1,n}$
	LINK$_{2,j-1}$	NODE$_{2,j}$	.	LINK$_{2,n-1}$	NODE$_{2,n}$
. .	LINK$_{3,j-1}$	NODE$_{3,j}$		LINK$_{3,n-1}$	NODE$_{3,n}$
			.		
	LINK$_{i,j-1}$	NODE$_{i,j}$		LINK$_{i,n-1}$	NODE$_{i,n}$
					:
	LINK$_{m,j-1}$	NODE$_{m,j-1}$		LINK$_{m,n-1}$	NODE$_{m,n}$

m = number of alternative parallel routes in set

n = number of nodes in an alternative

Figure 3 PARALLEL FACILITIES TABLE

referred to as set Z. The row's second column contains the number of alternatives comprising the set. For this example, column two is 3 (three). Column three contains a number identifying which alternative within the set is to be used during an EDB run as the selected traffic-bearing route. In the example, if the routine described in the second of the three rows describing the set is selected as the traffic-bearing route, column three contains a 2 (two), referring to the second of the three rows. The fourth column contains, possibly, a different value for each row in the table. The value gives the number of nodes (ports and non-port nodes) included in the route described by the row. Each alternative route begins, and ends, with a node identification number. Therefore, the minimum value possible for column four is 2 (two end nodes bounding a link). The row's columns after the fourth are determined by the number of links comprising the alternative parallel route described by the row. The total number of columns in a row is given by adding 3 (three) to twice the number of nodes in the parallel route (given in Column 4). As an illustrative example, a table describing a set of alternative parallel routes may be given as follows:

Z	3	2	2	N_1	L_1	N_2		
Z	3	2	3	N_1	L_2	N_3	L_3	N_2
Z	3	2	2	N_1	L_4	N_2		

The N's and L's refer to nodes and links respectively.

Notice first that, since there are three alternative routes in the set, there are three rows in the table. The order in which the rows are given is arbitrary and non-critical. The first column of each row contains Z, the set identification number. The number of alternative routes in the set, 3 is in the second column of each row. For a hypothetical EDB run, the alternative given in the second of the three rows is selected as the traffic-bearing route, so that column 3 of each row contains a 2 (two). The alternative described by the first row contains 2 (two) nodes, giving column 4 of row 1 a value of 2. Adding 3 (three) to twice the number of nodes, indicates that row 1 has a total of seven columns. The last three columns therefore give the identification numbers of the two nodes and the identification number of the link which they bound. The first node ("first" here refers to the node located at the upstream end of the channel) has the identification number N_1. This is given in column 5. Column 6 gives the identification number, L_1, of the link joining the route's two nodes. Column 7 contains N_2, the last link's identification number. The second row of the table is prepared in the same manner. Notice that the route contains 3 (three) nodes, given in column 4. The second row, therefore, has nine columns. Column 5 contains the first link's identification number, N_1 (note that all three alternative routes begin and end with the same nodes). Column 6 contains the identification number, L_2 of the link connecting the first node with the second node, N_3, whose identification number is given in column 7. Column 8 contains the identification number, L_3, of the link connecting N_3 with the last node, N_2, given in column 9. The third and final row of the table is prepared in the same manner.

The final table of data included with the input data stream is the ALTERNATIVE WIDE COEFFICIENTS which supplies coefficients for expected transit time functions. The table is optional and is included only with an EVENT LOG run simulating a configuration which contains alternative parallel routes. The data supplied in this table provide coefficients for factors summarized over an entire parallel route as opposed to the other coefficients which are concerned with individual links within a route.

B. OUTPUT DATA BASE

NETSIM/SHIP produces two different types of data bases. The first data base, the EDB, produces essentially two forms of data: first, that which gives prior conditions at a link before a vessel starts its transit and, second, that which gives the subsequent arrival and exit times for the vessel. These data are analyzed, using regression techniques, to derive functions describing the relationship between current conditions within a link and a vessel's subsequent transit time through that link. An EDB record is written for each link within a selected traffic-bearing route at the time a vessel arrives at one of the route's two boundary nodes. These records describe the currently existing conditions in the links. As a vessel moves through each of the links comprising the route, a second record is written for each link which gives the vessel's arrival time and its exit time. The exact format in which the records are written is adequately described in the User's Manual (see Appendix A) and, to prevent duplication, is not discussed here.

NETSIM/SHIP's second type of output data base, the EVENT LOG, consists of records that describe every event occurrence. The basic datum in the EVENT LOG records is a four-digit event code which encodes the exact sequence of events and conditions which leads to the occurrence of a particular event. The four digit codes, with their interpretations, and the format of the EVENT LOG records are detailed in the User's Manual.

III. MODEL STRUCTURE

The following discussion on NETSIM/SHIP's structural characteristics is broken into four subdivisions to correspond with each of the model's four logic modules. Each subdivision includes a detailed discussion on the logic that makes up that particular subdivision's module, as well as a summary flow chart of the logic process.

A. INITIALIZATION MODULE

The initialization of NETSIM/SHIP's values for a simulation run is accomplished through the ROUTINE FOR INITIALIZATION, the one routine comprising the initialization module. The routine's logic is straightforward, being concerned primarily with reading in the input data stream. An illustration of the routine's logic flow is given in Figure 4.

The initialization routine first reads in parameters needed by the model to arrange for certain options during the simulation run and to define the network configuration being simulated. These parameters, along with the rest of the input data stream, are discussed in detail in the Appendix.

The initialization module determines whether the network configuration being simulated contains any ports, lakes, reaches and/or locks. Then, for each port, lake, reach and/or lock in the configuration, the associated attributes of each entity are read in.

NETSIM/SHIP has two modes through which data describing the fleet to be simulated may be provided. The data provided through the two modes are basically identical. There are format variations, however, and these are described in detail in the Appendix. The first of the two modes provides the fleet data long with the regular input data stream. In this first mode,

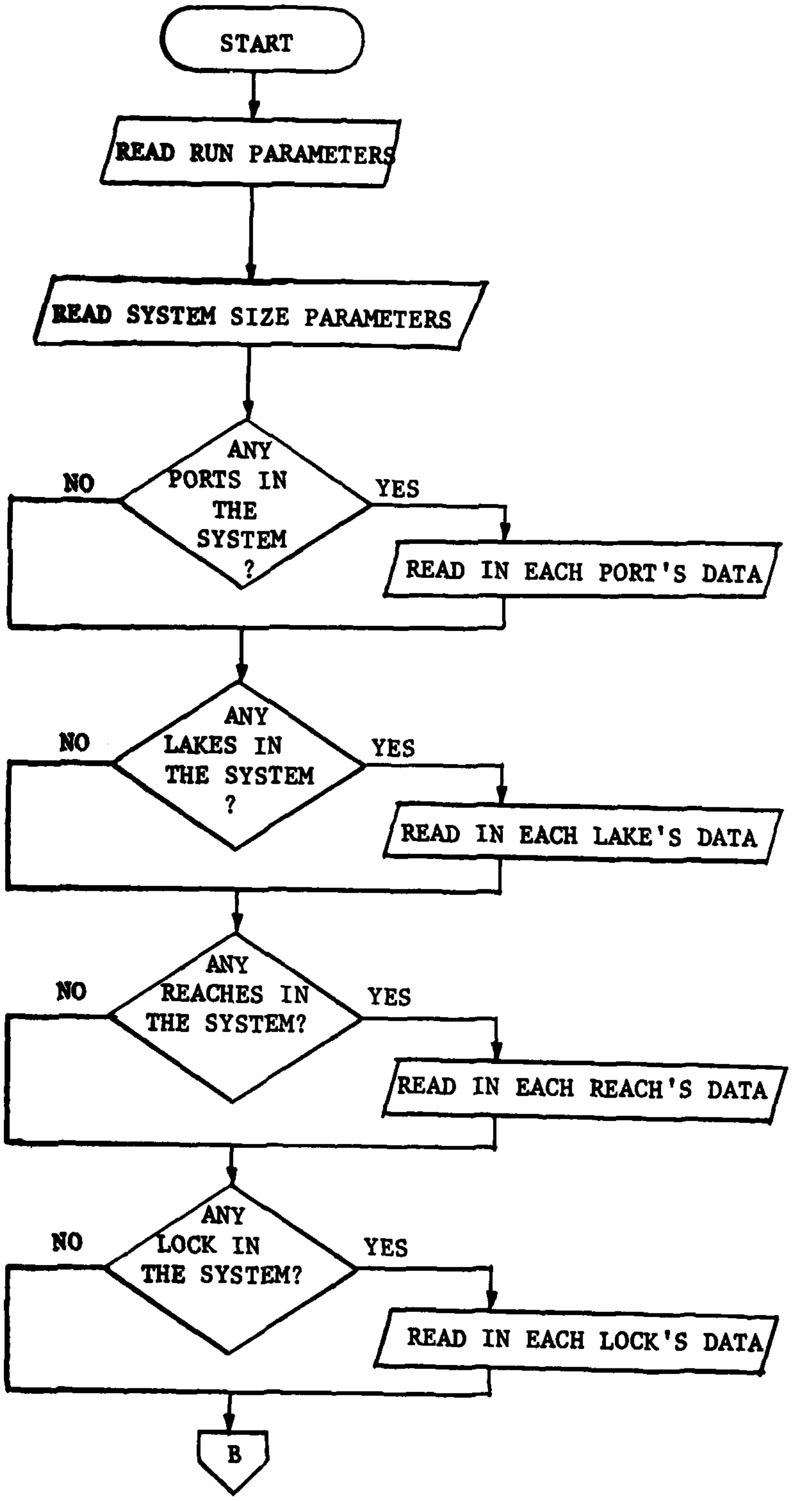


Figure 4. General Logic Flow of the ROUTINE FOR INITIALIZATION

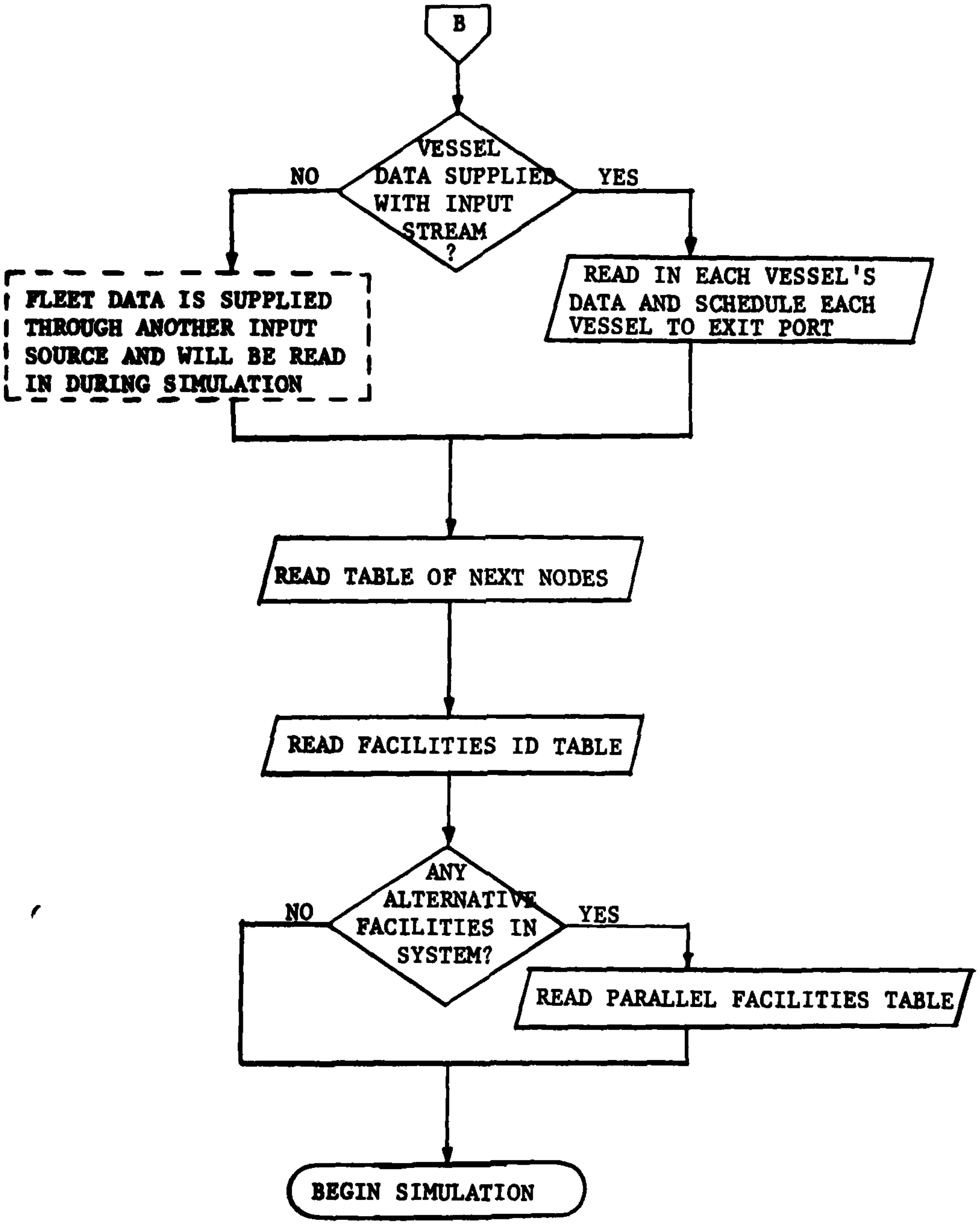

B
VESSEL DATA SUPPLIED WITH INPUT STREAM ?
NO
YES
FLEET DATA IS SUPPLIED THROUGH ANOTHER INPUT SOURCE AND WILL BE READ IN DURING SIMULATION
READ IN EACH VESSEL'S DATA AND SCHEDULE EACH VESSEL TO EXIT PORT
READ TABLE OF NEXT NODES
READ FACILITIES ID TABLE
ANY ALTERNATIVE FACILITIES IN SYSTEM?
NO
YES
READ PARALLEL FACILITIES TABLE
BEGIN SIMULATION

Figure 4. (Continued)

the initialization routine reads in each vessel's attribute data and assigns each vessel to its initial port for the beginning of the simulated shipping season. In the second mode, the fleet data are provided via an external input unit. The data are read in during simulation in order of the vessels' port exits.

Finally, the initialization routine reads in data tables that describe the network configuration--a tabular network map, in other words. These tables include the TABLE OF NEXT NODES, the FACILITIES ID TABLE, and, if the network configuration being simulated contains any parallel facilities, the PARALLEL FACILITIES TABLE.

B. TRAFFIC CONTROL MODULE

With all the necessary data read in, the actual simulation begins with the earliest scheduled port exit. The EVENT FOR PORT EXIT transfers program control to the traffic control module, where the ROUTINE MOVEMENT CONTROLLER and the ROUTINE ALTERNATIVE SELECTOR determine the vessel's route and assign the vessel to the first link (facility) on its route. This control transfer, as well as the other inter-module relationships within NETSIM/SHIP is illustrated in Figure 5.

As each vessel leaves its initial port at the beginning of the simulated shipping season, it is the traffic control module which determines the route to be followed to the vessel's destination port. As a vessel transits a link on its route, the traffic control module is invoked to determine the vessel's next link. The traffic control module then transfers program control to the appropriate link module. These back-and-forth control transfers, as illustrated by Figure 6, continue until the end of the simulation period.

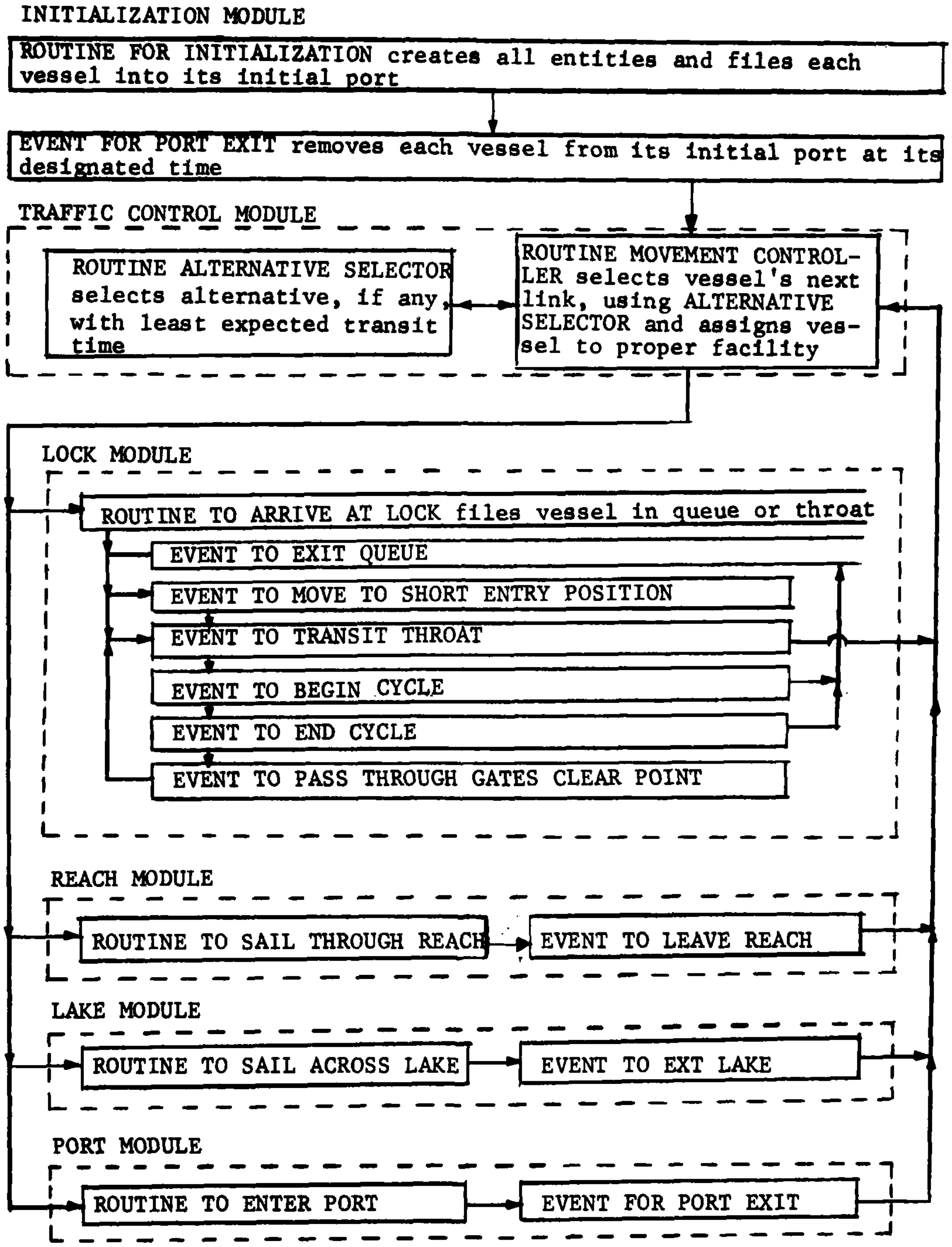

INITIALIZATION MODULE
ROUTINE FOR INITIALIZATION creates all entities and files each vessel into its initial port
EVENT FOR PORT EXIT removes each vessel from its initial port at its designated time
TRAFFIC CONTROL MODULE
ROUTINE ALTERNATIVE SELECTOR selects alternative, if any, with least expected transit time
ROUTINE MOVEMENT CONTROLLER selects vessel's next link, using ALTERNATIVE SELECTOR and assigns vessel to proper facility
LOCK MODULE
ROUTINE TO ARRIVE AT LOCK files vessel in queue or throat
EVENT TO EXIT QUEUE
EVENT TO MOVE TO SHORT ENTRY POSITION
EVENT TO TRANSIT THROAT
EVENT TO BEGIN CYCLE
EVENT TO END CYCLE
EVENT TO PASS THROUGH GATES CLEAR POINT
REACH MODULE
ROUTINE TO SAIL THROUGH REACH
EVENT TO LEAVE REACH
LAKE MODULE
ROUTINE TO SAIL ACROSS LAKE
EVENT TO EXT LAKE
PORT MODULE
ROUTINE TO ENTER PORT
EVENT FOR PORT EXIT

Figure 5. Interrelationships Among NETSIM/SHIP Modules

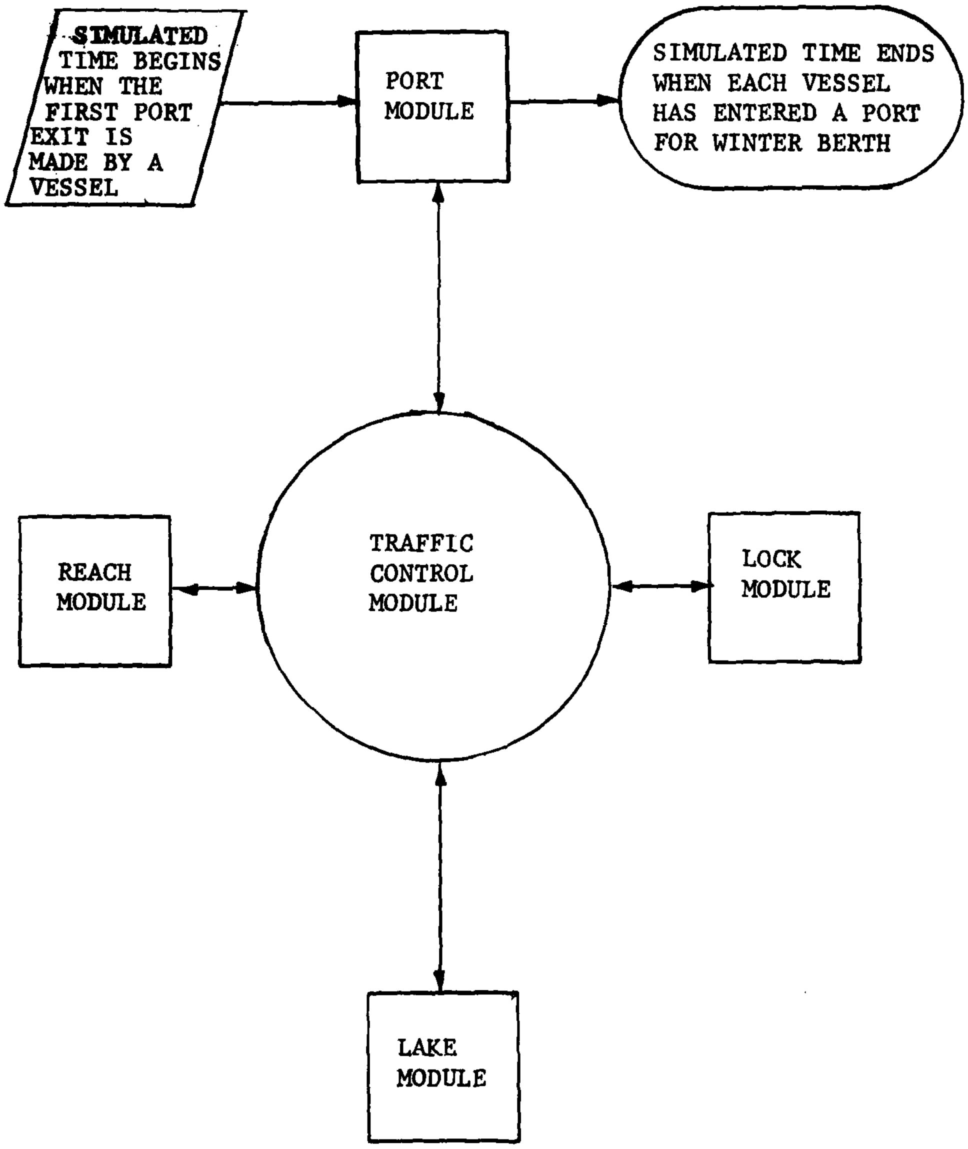

SIMULATED TIME BEGINS WHEN THE FIRST PORT EXIT IS MADE BY A VESSEL
PORT MODULE
SIMULATED TIME ENDS WHEN EACH VESSEL HAS ENTERED A PORT FOR WINTER BERTH
TRAFFIC CONTROL MODULE
REACH MODULE
LOCK MODULE
LAKE MODULE

Figure 6. Conceptual Structure of NETSIM/SHIP

1. THE ROUTINE MOVEMENT CONTROLLER

The MOVEMENT CONTROLLER routine has, as its basic function, the selection of a vessel's next link. It performs this function by accessing a vessel's map attributes which reveal a vessel's current location. Knowing a vessel's current location and its destination, the MOVEMENT CONTROLLER references the system map (made up of the TABLE OF NEXT NODES, the PARALLEL FACILITIES TABLE, and the FACILITIES ID TABLE) to determine the next link through which a vessel is to transit in order to reach its destination (refer to Figure 7).

Each vessel's map attributes are updated whenever the vessel invokes the MOVEMENT CONTROLLER. A vessel's map includes identification numbers for the following four nodes: (1) PREVIOUS NODE, (2) CURRENT NODE, (3) NEXT NODE, and (4) DESTINATION. A vessel's current location is always given by its previous, current, and next nodes. Its position is kept current by updating these three map attributes as the vessel progresses along its route. The node identification number stored as a vessel's PREVIOUS NODE is replaced with the identification number of the vessel's CURRENT NODE. The identification number of the CURRENT NODE is, in turn, replaced with that of the vessel's NEXT NODE. At this point in the map updating, it is possible to determine whether a vessel's current location is its destination port. If a vessel's CURRENT NODE, NEXT NODE, and DESTINATION all contain the same node identification number, the vessel has arrived at its destination port and the MOVEMENT CONTROLLER transfers program control to the port module. If the CURRENT NODE, NEXT NODE, and DESTINATION are not equal, the MOVEMENT CONTROLLER continues to determine a vessel's updated NEXT NODE.

The support module is invoked to search the TABLE OF NEXT NODES, one of the three tables comprising the system map. From the TABLE OF NEXT NODES, the MOVEMENT CONTROLLER learns the identification number of a vessel's next node,

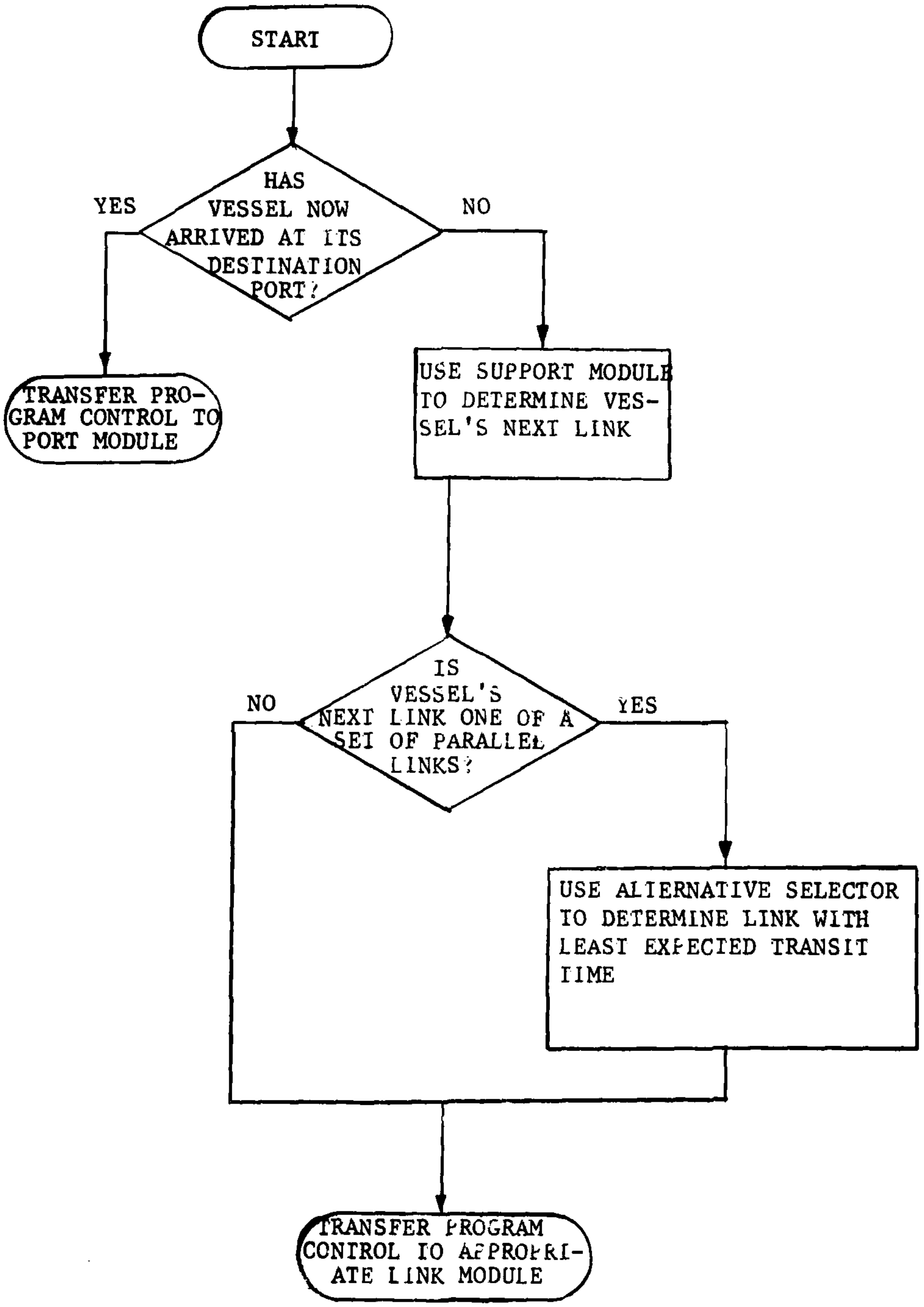

START
HAS VESSEL NOW ARRIVED AT ITS DESTINATION PORT?
YES
NO
TRANSFER PROGRAM CONTROL TO PORT MODULE
USE SUPPORT MODULE TO DETERMINE VESSEL'S NEXT LINK
IS VESSEL'S NEXT LINK ONE OF A SET OF PARALLEL LINKS?
NO
YES
USE ALTERNATIVE SELECTOR TO DETERMINE LINK WITH LEAST EXPECTED TRANSIT TIME
TRANSFER PROGRAM CONTROL TO APPROPRIATE LINK MODULE

Figure 7 General Logic Flow of the ROUTINE MOVEMENT CONTROLLER

as well as whether a vessel is faced with alternative parallel routes. If there are no parallel alternatives available to a vessel at its current location, the MOVEMENT CONTROLLER determines the identification number of the vessel's next link and transfers program control to the appropriate link module. If there are alternatives from which a vessel's next link may be selected, the MOVEMENT CONTROLLER, using the ALTERNATIVE SELECTOR routine, determines the one alternative offering the least expected transit time. From this alternative route with the least expected transit time, the MOVEMENT CONTROLLER selects a vessel's next link and transfers program control to the appropriate link module. Once a vessel is transiting a route which is one of a set of parallel alternatives, the MOVEMENT CONTROLLER directs its transit by referencing the PARALLEL FACILITIES TABLE, which contains the identification number of every node and link comprising an alternative route.

2. The ROUTINE ALTERNATIVE SELECTOR

The sole function of the ALTERNATIVE SELECTOR routine is to select one of a set of alternative parallel routes for a vessel to transit, using least expected transit time as the criterion for selection. The ALTERNATIVE SELECTOR first examines each alternative route to determine whether any of the routes contain a lock which is too small to accommodate the vessel (see Figure 8). If any alternative contains such a lock, that alternative is removed from consideration and the selection is made from the remaining routes.

The expected transit time for each available alternative is computed by the ALTERNATIVE SELECTOR using expected transit time functions which include weighted measures of existing traffic conditions within each link of the alternatives. The weights, or coefficients, for the factors included in the expected transit time functions are derived through regression analysis of

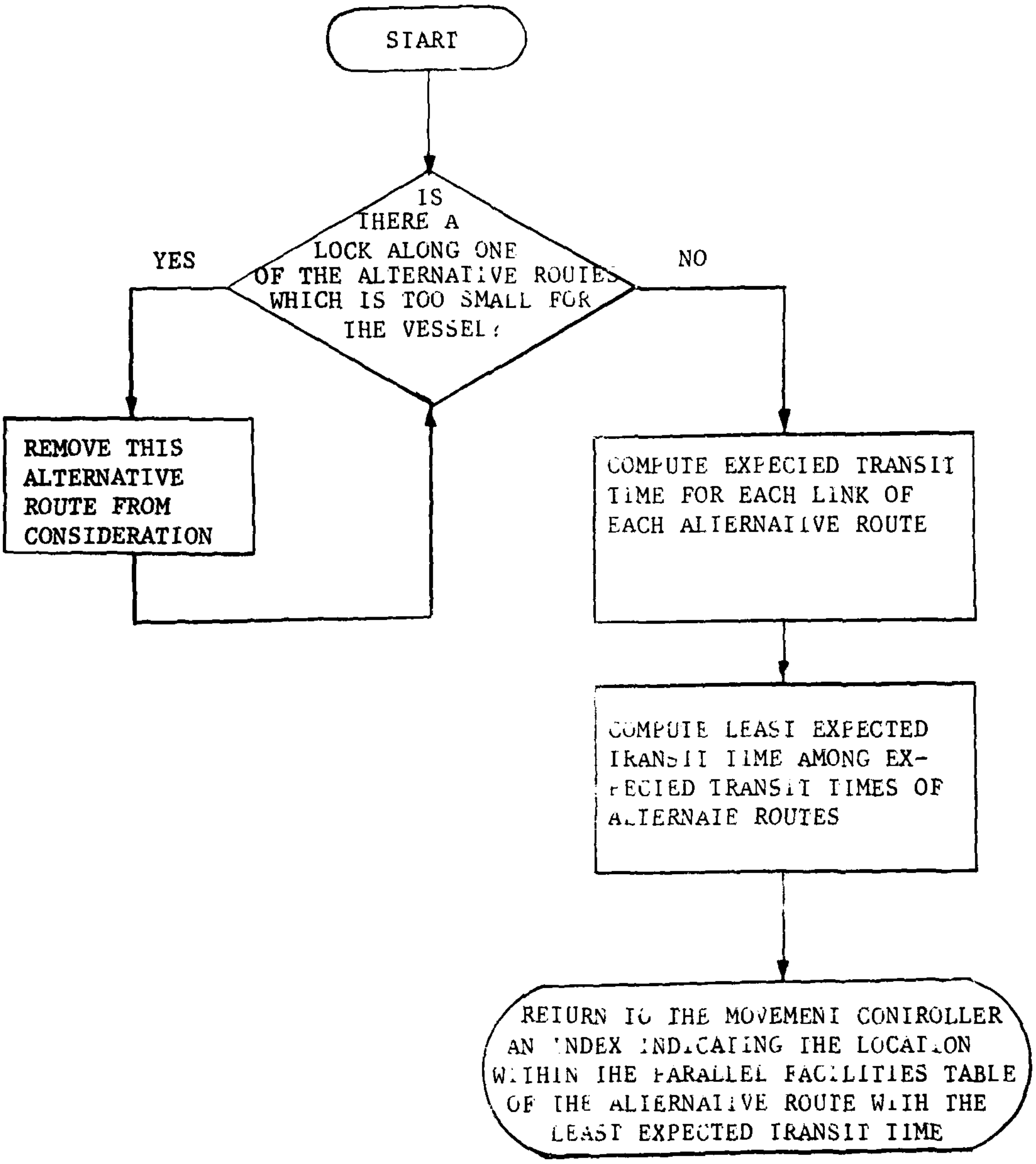

START
IS THERE A LOCK ALONG ONE OF THE ALTERNATIVE ROUTES WHICH IS TOO SMALL FOR THE VESSEL?
YES
NO
REMOVE THIS ALTERNATIVE ROUTE FROM CONSIDERATION
COMPUTE EXPECTED TRANSIT TIME FOR EACH LINK OF EACH ALTERNATIVE ROUTE
COMPUTE LEAST EXPECTED TRANSIT TIME AMONG EXPECTED TRANSIT TIMES OF ALTERNATE ROUTES
RETURN TO THE MOVEMENT CONTROLLER AN INDEX INDICATING THE LOCATION WITHIN THE PARALLEL FACILITIES TABLE OF THE ALTERNATIVE ROUTE WITH THE LEAST EXPECTED TRANSIT TIME

Figure 8. General Logic Flow of the ROUTINE ALTERNATIVE SELECTOR

the Experience Data Base (EDB). The EDB is generated by a series of preliminary simulation runs wherein all transits are directed through selected alternative parallel routes. The expected transit time functions also take transit direction into consideration since the impact upon expected transit time of existing link traffic conditions is dependent upon that traffic's proximity to the subject vessel.

After computing an expected transit time for each available alternative parallel route, the ALTERNATIVE SELECTOR then chooses that alternative with the least expected transit time as the one to be transited. An index, indicating the location within the PARALLEL FACILITIES TABLE of the description of the selected alternative, is returned to the MOVEMENT CONTROLLER, along with program control. The MOVEMENT CONTROLLER, now knowing which alternative will yield the least expected transit time, selects the vessel's first link on that alternative route and transfers program control to the appropriate link module.

C. LINK MODULE(s)

The link module may be a single link model or it may contain many such models. The quantity depends upon the particular application. The current NETSIM/SHIP implementation includes the following three link models: (1) lock, (2) reach, and (3) lake. Each of these three link models, along with NETSIM/SHIP's port model and the vessel entity, are discussed below.

1. LOCK: a Link Module

For obvious reasons, the lock module is considerably more complicated than the reach and lake modules. NETSIM/SHIP's lock module builds upon the module developed for the NCDD model, adding, among others, a service look-ahead feature. The service look-ahead feature gives the lock model the capability to commit the lock to one of two vessels based upon an expected arrival time in the chamber for each vessel Basically, this feature simulates the decision process of an experienced rational lockmaster who, seeing a vessel approaching his lock from either end, must decide whether it is better to recycle the chamber while empty if necessary to serve one vessel, or to leave the chamber as is to serve the other vessel. NETSIM/SHIP's implementation of the service look-ahead feature is based upon SIMSCRIPT's FOR statement which allows access to information concerning upcoming events. With the FOR statement, NETSIM/SHIP can determine whether an arrival at the lock is imminent and the exact time at which the arrival is to occur Given this information, the model can determine which committment would tend to minimize a vessel's transit time

A similar feature included in NETSIM/SHIP's lock model simulates an experienced rational lockmaster's decision process in deciding whether to recycle the chamber empty to receive a vessel which is approaching an idle lock. This feature is referred to as the recycle look-ahead feature The model determines whether, upon a vessel's arrival at the idle lock, the chamber has been idle long enough to have recycled before the vessel arrived. If it has been idle long enough, the model assumes that an experienced lockmaster would have recycled having seen the approaching vessel, and instantaneously recycles the chamber in order to simulate the lockmaster's action.

The lock model includes nine distributions for describing time delays. Seven of the first nine distributions are direction-differentiated, so that there are actually sixteen time distributions used in the model Figure 9 describes the nine time elements in relation to the physical layout of the lock and Table 1 provides a detailed description The nine time elements are used throughout seven different logic routines that comprise NETSIM/SHIP's lock module. Each of the seven routines is discussed below

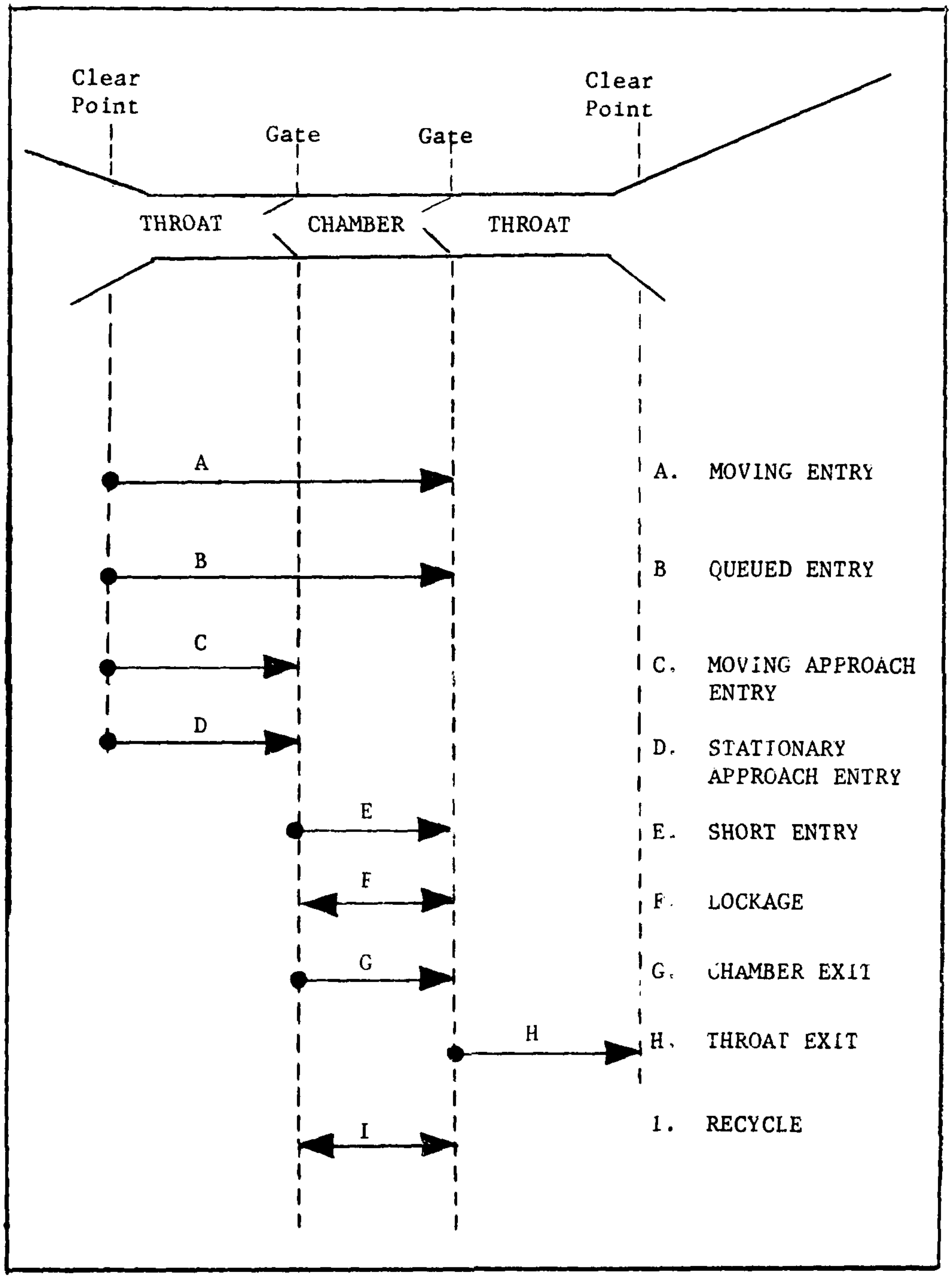


Figure 9. Schematic of Simulated Locking Time Events

TABLE 1. DESCRIPTION OF LOCKING TIME EVENTS

EVENT NAME	DESCRIPTION
A. MOVING ENTRY	into the lock chamber from the Clear Point at the end of the entry throat.
Begins:	when the bow of the ship passes the Clear Point
Ends:	when the gates begin to close astern of the ship
B. QUEUED ENTRY	into the lock chamber from the head of the queue adjacent to the Clear Point.
Begins:	when the gates are fully open and the chamber is free
Ends:	when the gates begin to close astern of the ship
C. MOVING APPROACH ENTRY	from the Clear Point to a position in the entry throat just clear of the entry gate.
Begins:	when the bow of the ship passes the Clear Point
Ends:	when the ship comes to rest in the entry throat
D. STATIONARY APPROACH ENTRY	from the head of the queue adjacent to the Clear Point to a position in the entry throat just clear of the entry gate.
Begins:	when the bow of the ship passes the Clear Point
Ends:	when the ship comes to rest in the entry throat
E. SHORT ENTRY	into the lock chamber from a stationary position just clear of the entry gate in the entry throat.
Begins:	when the gates are fully open and the chamber is free
Ends:	when the gates begin to close astern of the ship
F. LOCKAGE	of a ship at rest in the chamber
Begins:	when the entry gates begin to close astern of the ship
Ends:	when the exit gates are fully opened after the change in water level
G. CHAMBER EXIT	from chamber to position where the ship's stern clears the exit gate.
Begins:	when the exit gates are fully opened after the change in water level
Ends:	when the ship's stern is clear of the exit gate
H. THROAT EXIT	from position where the ship's stern clears the exit gate to the Clear Point at the end of the exit throat.
Begins:	when the ship's stern is clear of the exit gate
Ends:	when the stern of the ship passes the Clear Point
I. RECYCLE	of the water level with no ship in the chamber
Begins:	when the gates begin to close
Ends:	when the opposite gates are fully open to receive an incoming ship

a. ROUTINE TO ARRIVE AT LOCK

There are basically three movements that can occur upon a vessel's arrival at a lock. The arriving vessel can (1) enter the near queue, (2) begin moving into the short-entry position, or (3) begin moving into the chamber. The movement made by a particular vessel depends upon current lock traffic conditions. Determining the current status of these conditions occupies the major portion of the ARRIVE AT LOCK routine's logic. A summarized flow chart of the logic is given in Figure 10.

If the lock's near queue is occupied, an arriving vessel automatically enters that queue. If, however, the near queue is empty, the near throat is checked. If there is a vessel here, regardless of direction of transit, the arrival vessel joins the near queue. If the near throat is also empty, the logic checks the status of the chamber. The chamber's status can be either (1) empty, (2) contains a vessel moving in the same direction, or (3) contains a vessel moving in the opposite direction. If the chamber is empty, only the water level remains to be checked. If the water level in the chamber is properly adjusted, the arriving vessel begins moving into the chamber. If the water level is improper upon a vessel's arrival, further checking is required in order to make the decision whether to recycle the chamber empty. If the lock's far throat is empty, the lock's far queue is also checked. If the far queue is also empty, the lock model's service look-ahead feature is invoked (Actually, this feature can be included or omitted according to the user's wishes. Discussion here assumes its use.) to check the next reach for an imminent arrival. If there is an imminent arrival from the next reach, the service look-ahead logic compares the times at which the two vessels could be expected to be in tie-up position inside the chamber. The lock is committed to that vessel with the earlier

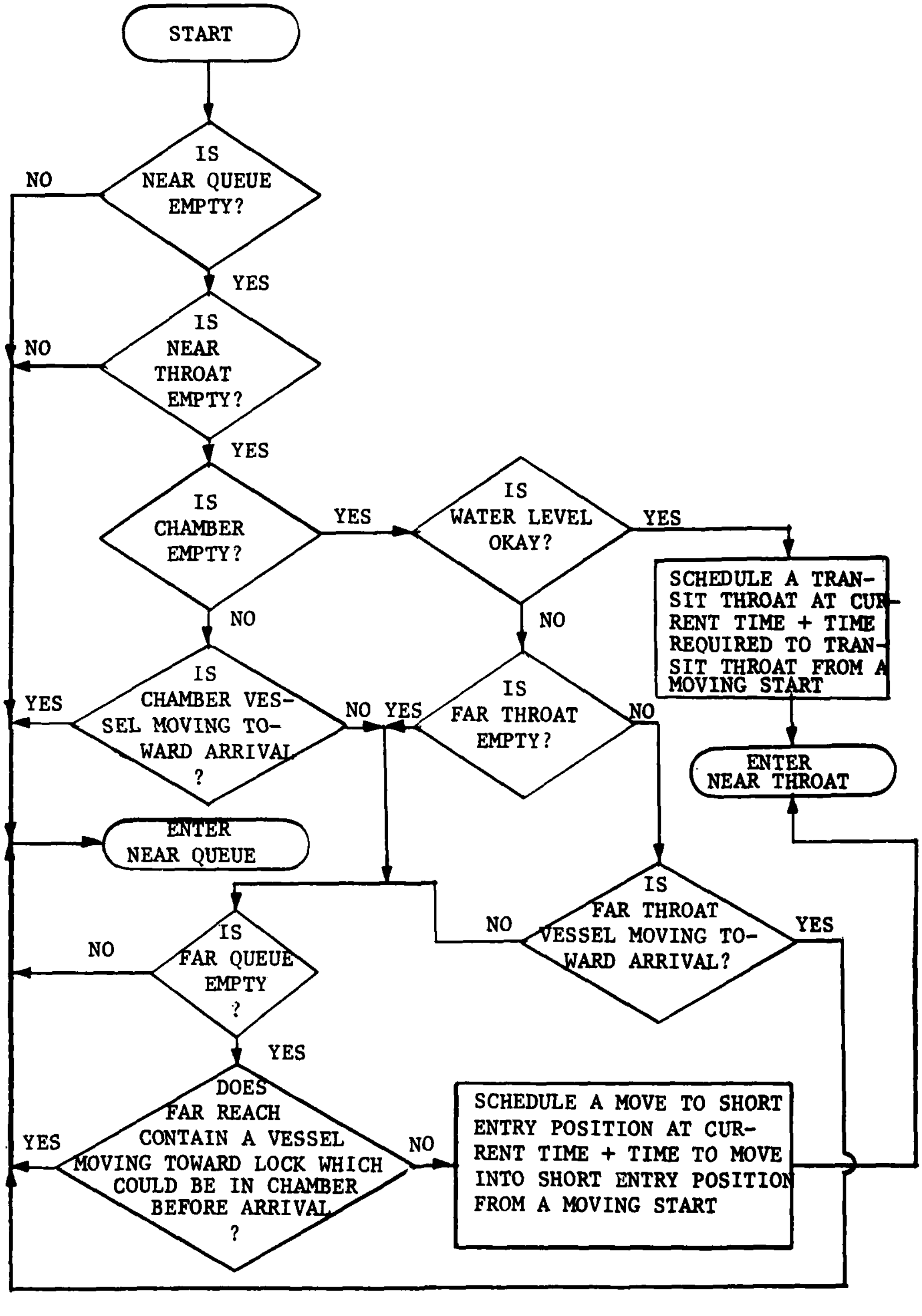

START
IS NEAR QUEUE EMPTY?
NO
YES
IS NEAR THROAT EMPTY?
NO
YES
IS CHAMBER EMPTY?
YES
IS WATER LEVEL OKAY?
YES
SCHEDULE A TRANSIT THROAT AT CURRENT TIME + TIME REQUIRED TO TRANSIT THROAT FROM A MOVING START
ENTER NEAR THROAT
NO
NO
IS CHAMBER VESSEL MOVING TOWARD ARRIVAL ?
YES
NO YES
IS FAR THROAT EMPTY?
NO
ENTER NEAR QUEUE
IS FAR THROAT VESSEL MOVING TOWARD ARRIVAL?
NO
YES
IS FAR QUEUE EMPTY ?
NO
YES
DOES FAR REACH CONTAIN A VESSEL MOVING TOWARD LOCK WHICH COULD BE IN CHAMBER BEFORE ARRIVAL ?
YES
NO
SCHEDULE A MOVE TO SHORT ENTRY POSITION AT CURRENT TIME + TIME TO MOVE INTO SHORT ENTRY POSITION FROM A MOVING START

Figure 10. General Logic Flow of the ROUTINE TO ARRIVE AT LOCK

expected tie-up time, beginning a recycle immediately if the current arrival vessel is expected to be earlier. If the approaching vessel from the next reach has the earlier expected tie-up time, the current arrival vessel enters the near queue. If there is no imminent arrival in the next reach, the chamber begins recycling immediately and the current arrival vessel moves into its entry throat, either into the short-entry position or directly into the chamber, depending upon the chamber recycle time. If the lock's far queue is not empty, the arrival vessel moves into the near queue without making use of the service look-ahead feature. If the far throat contains a vessel moving toward the current arrival vessel, i.e., into the chamber, the current arrival vessel joins the near queue. If the vessel in the far throat is moving in the same direction as the current arrival vessel, the logic checks the far queue, etc., as described above.

b. EVENT TO EXIT QUEUE

When traffic conditions are such that a vessel, which is currently in a queue, can leave that queue, there are two movements possible for the vessel. It is the function of the EVENT TO EXIT QUEUE to investigate lock traffic conditions and to schedule the occurrence of one of the two possible movements. The logic for this investigation is illustrated in Figure 11. If the chamber is empty and the water level is proper, the vessel can move from its queue directly into the chamber. If the chamber is empty but the water is not at the proper level, the logic must determine whether the chamber is currently recycling or is currently scheduled to begin recycling. If the chamber is neither currently recycling nor is currently scheduled to begin recycling, an error has occurred. The logic of NETSIM/SHIP's lock module is designed in such a way that a vessel will

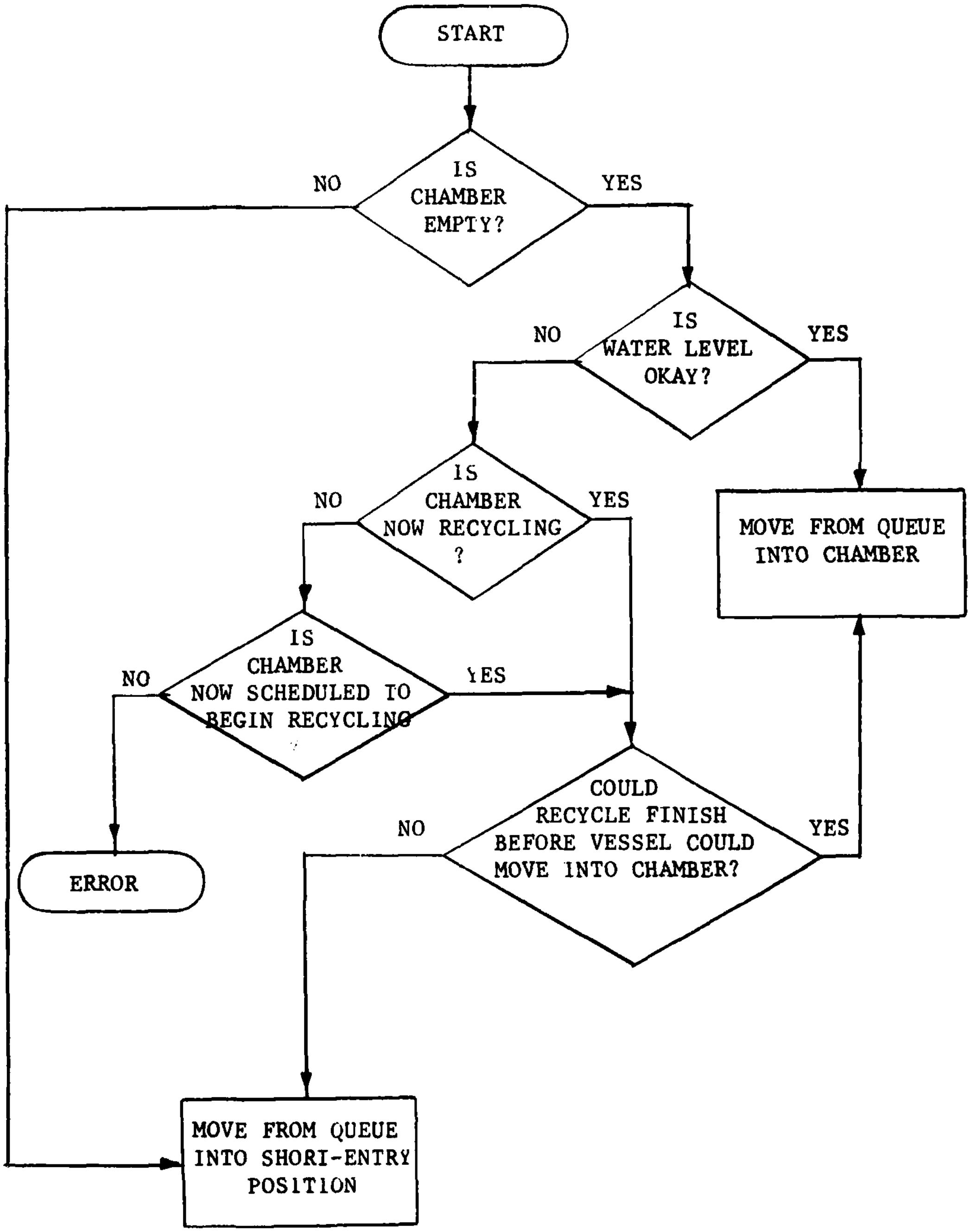

START
IS CHAMBER EMPTY?
NO
YES
IS WATER LEVEL OKAY?
NO
YES
IS CHAMBER NOW RECYCLING ?
NO
YES
MOVE FROM QUEUE INTO CHAMBER
IS CHAMBER NOW SCHEDULED TO BEGIN RECYCLING
NO
YES
COULD RECYCLE FINISH BEFORE VESSEL COULD MOVE INTO CHAMBER?
NO
YES
ERROR
MOVE FROM QUEUE INTO SHORT-ENTRY POSITION

Figure 11. General Logic Flow of the EVENT TO EXIT QUEUE

never exit a queue when the chamber is empty and the water is at the wrong level unless the chamber is already recycling or is about to begin recycling. In the case where a vessel exits a queue and there is a vessel in the chamber, the former queued vessel moves into the short-entry position.

c. EVENT TO MOVE TO SHORT-ENTRY POSITION

The EVENT TO MOVE TO SHORT-ENTRY POSITION is included in the lock module to allow for the situations that permit a vessel to enter a throat before the chamber is ready to receive it. The logic that determines when such a situation exists is contained in the other routines of the module, so that the MOVE TO SHORT-ENTRY POSITION logic is negligible. The event is necessary as a separate routine in order to provide a record of the short-entry movements made during simulation

d EVENT TO TRANSIT THROAT

The EVENT TO TRANSIT THROAT actually consists of two subroutines. Since a lock has two throats--an entry throat and an exit throat for each vessel--the logic describing the two different transits are different. The first throat transited by a vessel brings the vessel into the lock chamber, while the second throat transit brings a vessel to the lock's clear point, where the network's next link begins.

For the situation where a vessel is transiting its entry (or first) throat of a lock, the TRANSIT THROAT logic is relatively simple. As illustrated in Figure 12, the vessel moves into its tie-up position inside the lock and the chamber begins its process. On the other hand, the logic for a vessel's second throat transit is quite complicated.

The reason for the complexity in the logic used to simulate a vessel's exit from a lock is that a decision must be made as to whether to admit a vessel into the throat from which the current vessel is now leaving.

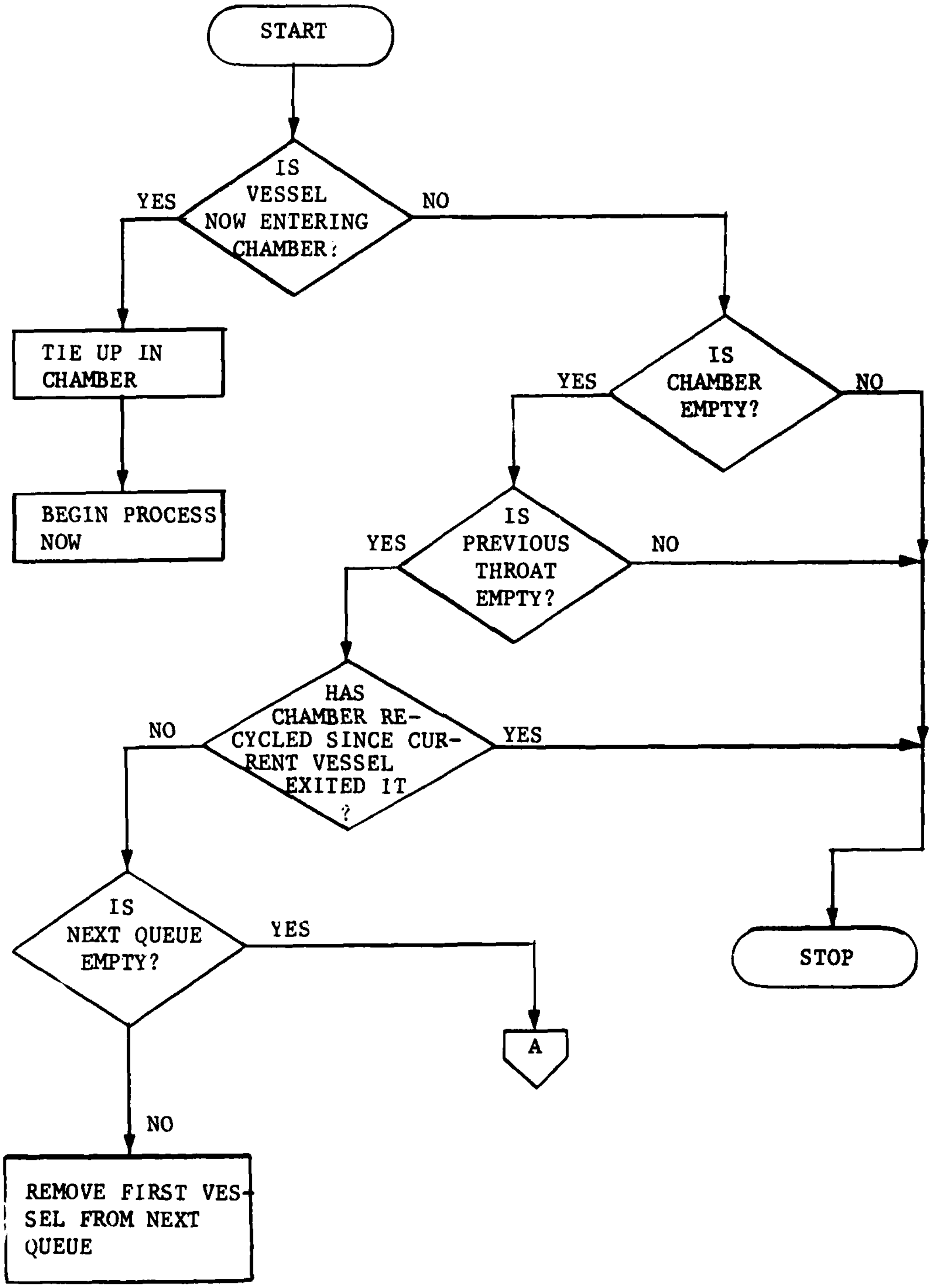

START
IS VESSEL NOW ENTERING CHAMBER?
YES
NO
TIE UP IN CHAMBER
BEGIN PROCESS NOW
IS CHAMBER EMPTY?
YES
NO
IS PREVIOUS THROAT EMPTY?
YES
NO
HAS CHAMBER RE-CYCLED SINCE CUR-RENT VESSEL EXITED IT ?
NO
YES
IS NEXT QUEUE EMPTY?
YES
NO
A
STOP
REMOVE FIRST VES-SEL FROM NEXT QUEUE

Figure 12. General Logic Flow of the EVENT TO TRANSIT THROAT

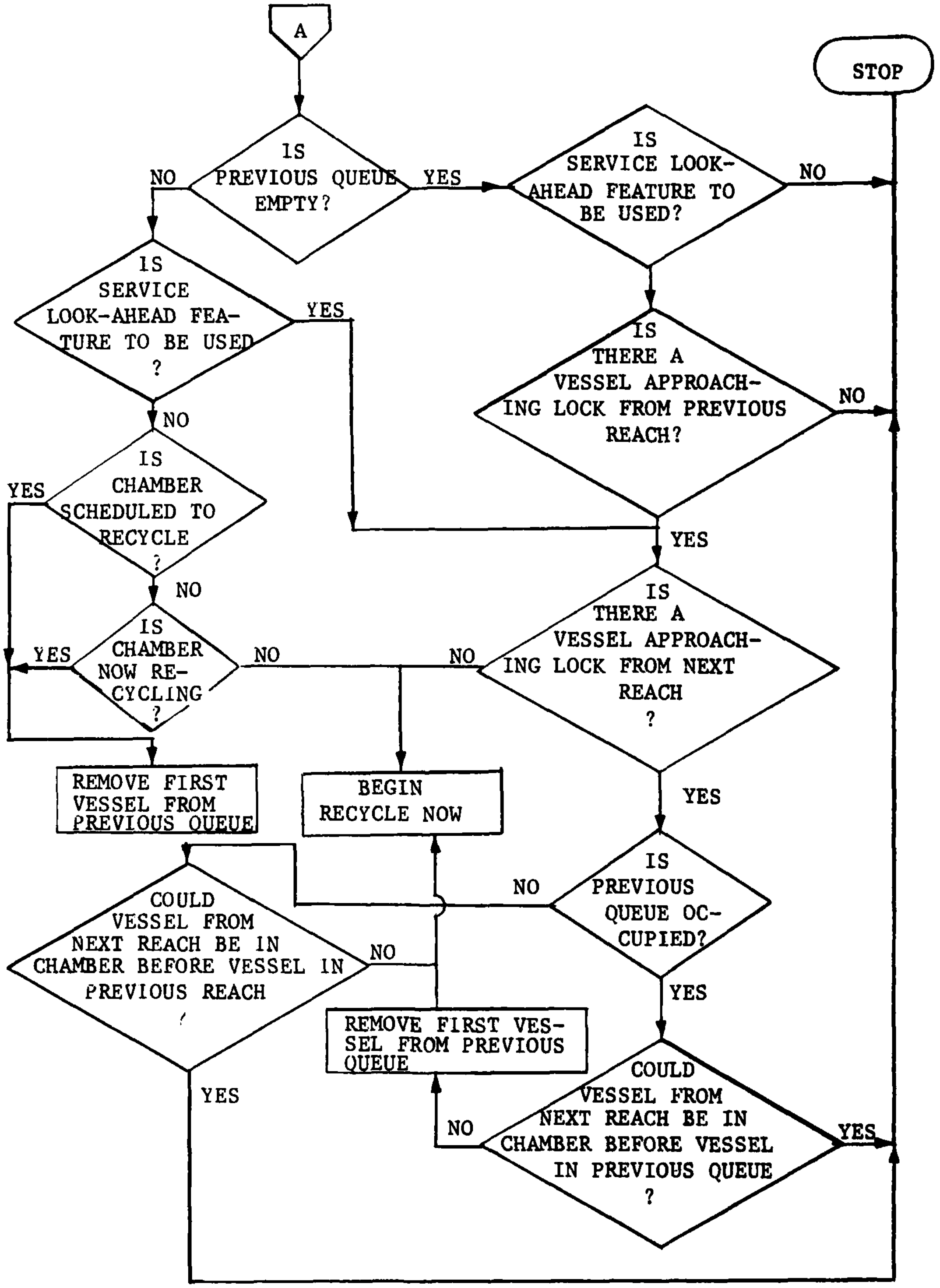

A
STOP
IS PREVIOUS QUEUE EMPTY?
NO
YES
IS SERVICE LOOK-AHEAD FEATURE TO BE USED?
NO
IS SERVICE LOOK-AHEAD FEA-TURE TO BE USED ?
YES
IS THERE A VESSEL APPROACH-ING LOCK FROM PREVIOUS REACH?
NO
NO
IS CHAMBER SCHEDULED TO RECYCLE ?
YES
YES
NO
IS THERE A VESSEL APPROACH-ING LOCK FROM NEXT REACH ?
IS CHAMBER NOW RE-CYCLING ?
YES
NO
NO
REMOVE FIRST VESSEL FROM PREVIOUS QUEUE
BEGIN RECYCLE NOW
YES
IS PREVIOUS QUEUE OC-CUPIED?
NO
COULD VESSEL FROM NEXT REACH BE IN CHAMBER BEFORE VESSEL IN PREVIOUS REACH ?
NO
YES
REMOVE FIRST VES-SEL FROM PREVIOUS QUEUE
COULD VESSEL FROM NEXT REACH BE IN CHAMBER BEFORE VESSEL IN PREVIOUS QUEUE ?
NO
YES
YES

Figure 12. (Continued)

Figure 12 also illustrates this process. If the lock's chamber contains a vessel, the chamber must have recycled empty to receive the vessel. This leaves nothing for the TRANSIT THROAT logic to do, since the lockmaster logic has already set other events in motion. If, however, the chamber is empty, the TRANSIT THROAT logic determines whether the previous throat is empty. If the previous throat contains a vessel, it is known that the lock has been committed to a second vessel moving sequentially through the lock. If the previous throat is empty, the lock may still be committed to a second sequential vessel. The logic determines whether the chamber has recycled since the current vessel left it. If it has, it must be awaiting an imminent arrival from the previous reach. If a recycle has not occurred, the logic now determines whether the next queue is occupied. If this queue is occupied, the TRANSIT THROAT logic arranges for the first vessel in the queue to exit that queue and begin moving into the chamber (this constitutes an alternating, or SOQA[2] queue discipline). However, if the next queue is empty, the logic checks the previous queue for a vessel that may move sequentially through the lock (this would constitute a FCFS[3] queue discipline). If the previous queue is empty and the service look-ahead feature of NETSIM/SHIP is not to be used, TRANSIT THROAT's work is finished. The lock is idle until the next vessel arrives. If the service look-ahead feature is to be used, the previous reach is checked for an imminent arrival. If there is no imminent arrival in the previous reach, the lock remains idle until its next arrival. If there is an approaching vessel in the previous reach, the service look-ahead feature then checks the next reach for an imminent arrival. If the next reach has no approaching vessel, the TRANSIT THROAT logic begins to

[2] Serve opposing queues alternatively.

[3] First come first served.

recycle the chamber in order to receive the vessel from the previous reach upon its arrival However, if the next reach does contain an approaching vessel, the service look-ahead feature determines which of the two imminent arrivals could move into the chamber earlier. The lock is committed to that vessel with the earlier expected chamber tie-up time, recycling the chamber if the vessel in the previous reach could be earlier.

If the previous queue is occupied, and NETSIM/SHIP's service look-ahead feature is not to be used, the TRANSIT THROAT logic determines whether the chamber is about to begin recycling to receive the vessel currently waiting in the previous queue. If a recycle is imminent, the first vessel in the previous queue exits that queue and moves into the previous throat. If no recycle is about to occur, it is determined whether the chamber is currently recycling. If a recycle is occurring, the first vessel in the previous queue enters the previous throat. If no recycle is currently occurring, one is begun.

In the situation where the previous queue is occupied and the model's service look-ahead feature is used, the next reach is checked for an imminent arrival. If there is a vessel approaching the lock from the next reach, its expected chamber tie-up time is compared with that of the first vessel in the previous queue The lock is committed to the vessel with the earlier expected tie-up time, recycling the chamber and removing the first vessel from the previous queue if that vessel is expected to be first. If there is no vessel approaching the lock from the next reach, the chamber is recycled to receive the first vessel in the previous queue.

e EVENT TO BEGIN CYCLE

The EVENT TO BEGIN CYCLE, like the EVENT TO TRANSIT THROAT, has a dual function, as illustrated in Figure 13. A chamber cycle can be an empty cycle (referred to as a recycle) or it may contain a vessel (referred to as a process). For the recycle, the EVENT TO BEGIN CYCLE does nothing more than schedule the end of the recycle at some randomly determined future time The logic concerned with an occupied cycle, i.e., a process, is more complex.

If the next queue by which the chamber vessel will pass is occupied, the BEGIN CYCLE logic only schedules the end of the process to occur. If, however, this queue is empty, the previous queue is checked to determine whether it is occuped If there is no vessel in the previous queue, the end of the process is scheduled. There is no need to check adjacent reaches at this point because other routines within the lock module contain the necessary logic to cover the possible situations. If the previous queue contains a vessel and the service look-ahead feature is not to be used, the end of the process is scheduled and an EVENT TO EXIT QUEUE is scheduled to remove the first vessel from the previous queue. If the service look-ahead feature is used and discovers that there is no vessel approaching the lock from the next reach, the same events occur, i.e., the end of the process is scheduled and the first vessel is removed from the previous queue. If the next reach does contain an imminent arrival, it is determined whether this arrival could be expected to be in its chamber tie-up position before the vessel from the previous queue could do so. The lock is committed to the vessel with the earlier expected arrival time, scheduling an EXIT QUEUE event if the vessel in the previous queue is expected to be earlier.

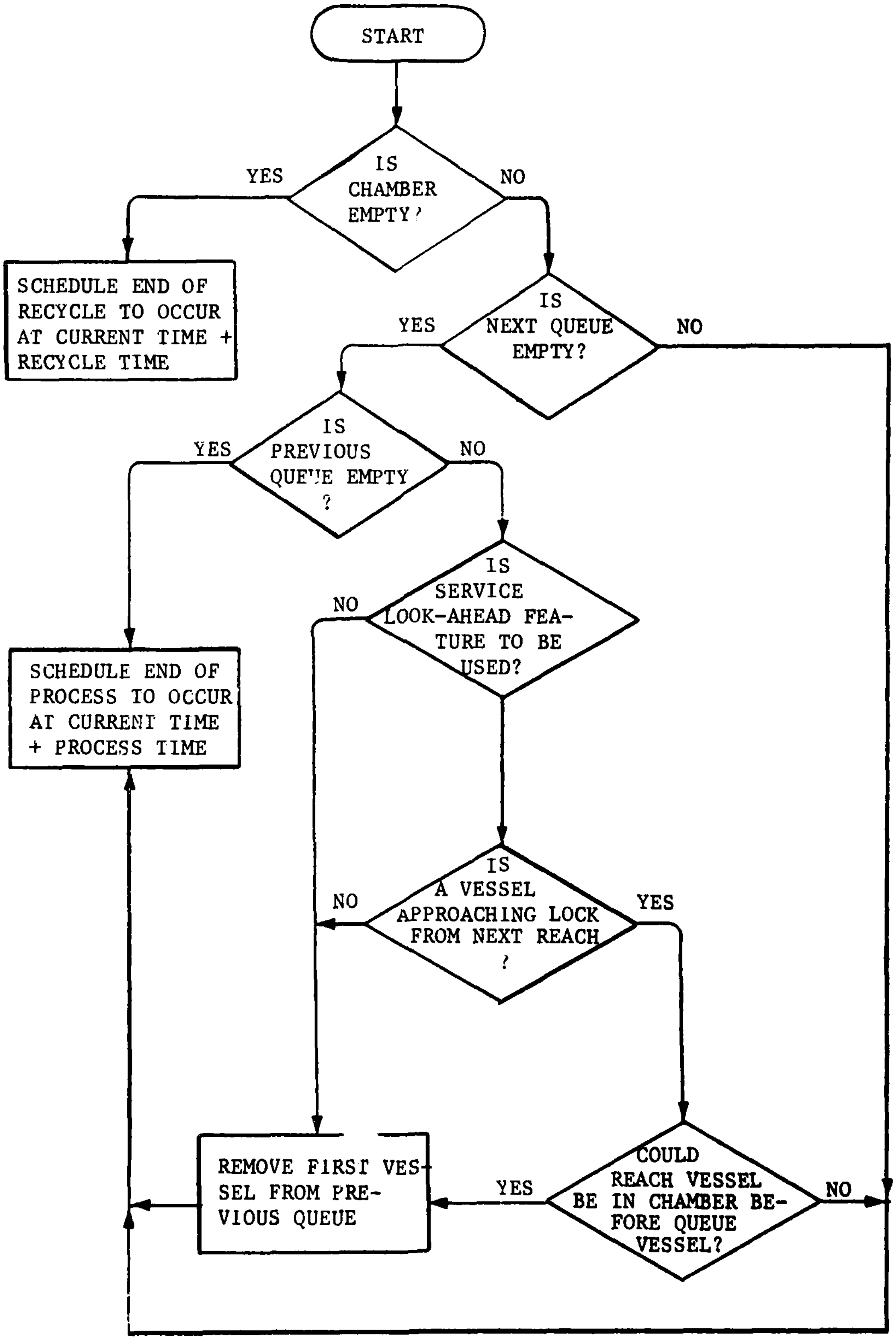

START
IS CHAMBER EMPTY?
YES
NO
SCHEDULE END OF RECYCLE TO OCCUR AT CURRENT TIME + RECYCLE TIME
IS NEXT QUEUE EMPTY?
YES
NO
IS PREVIOUS QUEUE EMPTY ?
YES
NO
IS SERVICE LOOK-AHEAD FEATURE TO BE USED?
NO
SCHEDULE END OF PROCESS TO OCCUR AT CURRENT TIME + PROCESS TIME
IS A VESSEL APPROACHING LOCK FROM NEXT REACH ?
NO
YES
REMOVE FIRST VESSEL FROM PREVIOUS QUEUE
YES
COULD REACH VESSEL BE IN CHAMBER BEFORE QUEUE VESSEL?
NO

Figure 13 General Logic Flow of the EVENT TO BEGIN CYCLE

f EVENT TO END CYCLE

When a lock chamber has completed a cycle, there are basically two sets of events that may occur Which set of events occurs depends on the current status of the chamber, i.e., whether the chamber contains a vessel or whether it is empty. These two event possibilities are illustrated in Figure 14.

If, at the end of a cycle, there is a vessel inside the chamber, the EVENT TO END CYCLE schedules the vessel to begin moving out of the chamber to a point in its exit throat where the vessel just clears the chamber gates. No other action is necessary since the lock model's "experienced lockmaster" logic mechanisms are associated with the EVENT TO PASS THROUGH GATES CLEAR POINT.

When the lock chamber is empty, indicating that the chamber has recycled to receive a waiting, or an approaching vessel, the END CYCLE logic determines which event must occur next in order to begin processing the next vessel. First, the throat which is currently at chamber water level is checked for a waiting vessel. If there is a vessel waiting in the throat's short-entry position, END CYCLE schedules a TRANSIT THROAT event to occur in order to move the vessel into tie-up position in the chamber If the vessel is not waiting in short-entry position, it must already be scheduled to move directly into the chamber. In this case, no further logic is required, since a TRANSIT THROAT event is already scheduled for moving the throat vessel into the chamber.

When the lock chamber and the water-level throat are empty, the END CYCLE logic determines whether the water-level queue contains a vessel which should move into the chamber for processing. If there is a vessel waiting in queue, an EVENT TO EXIT QUEUE is scheduled to occur immediately.

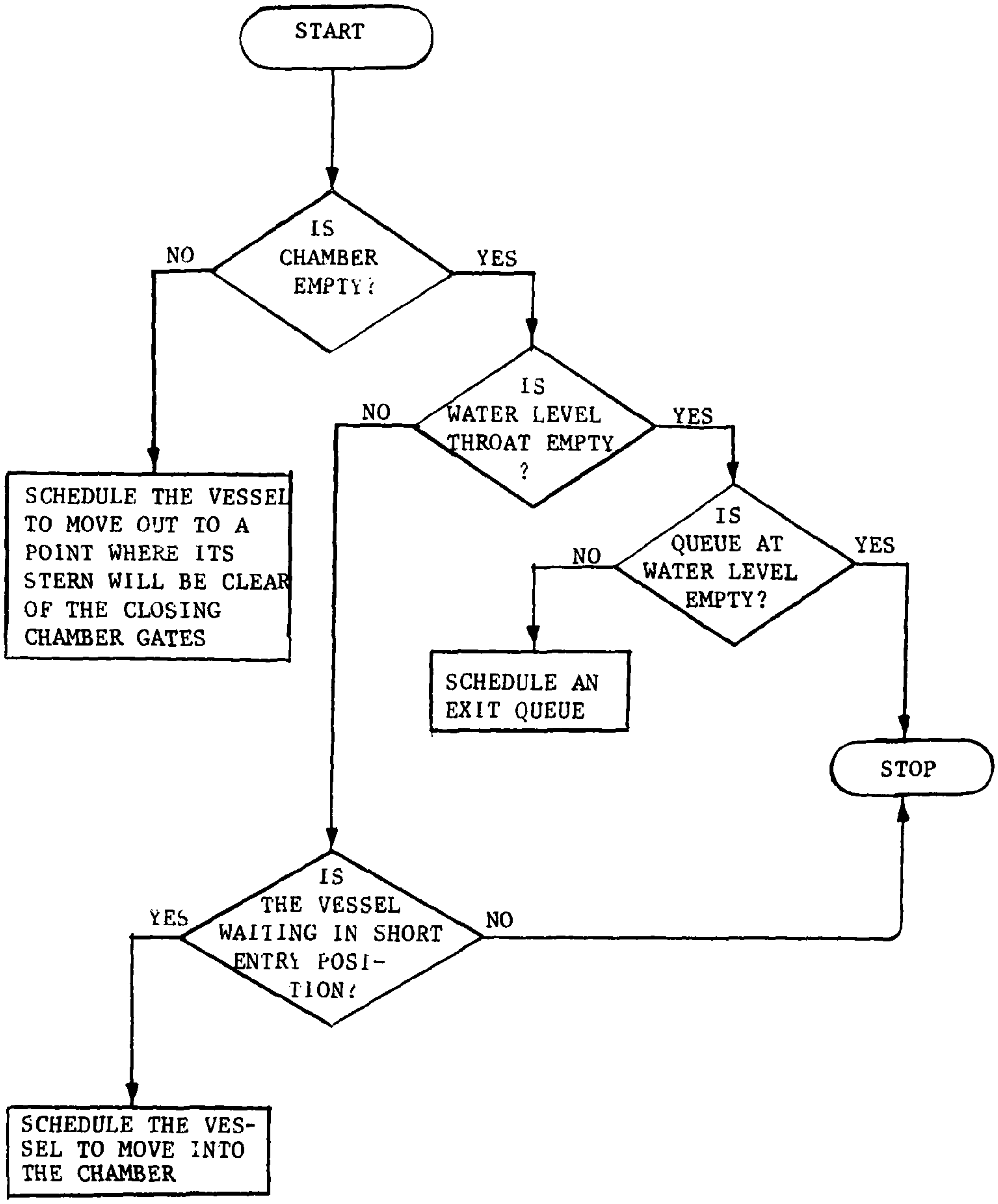

START
IS CHAMBER EMPTY?
NO
YES
IS WATER LEVEL THROAT EMPTY ?
NO
YES
SCHEDULE THE VESSEL TO MOVE OUT TO A POINT WHERE ITS STERN WILL BE CLEAR OF THE CLOSING CHAMBER GATES
IS QUEUE AT WATER LEVEL EMPTY?
NO
YES
SCHEDULE AN EXIT QUEUE
STOP
IS THE VESSEL WAITING IN SHORT ENTRY POSI- TION?
YES
NO
SCHEDULE THE VES- SEL TO MOVE INTO THE CHAMBER

Figure 14. General Logic Flow of the EVENT TO END CYCLE

However, if the water-level queue is empty, no further logic is necessary because the NETSIM/SHIP logic design is such that the only remaining possibility is that of an imminent arrival from the adjacent reach. This situation is handled in the ROUTINE TO ARRIVE AT LOCK, so that END CYCLE need continue no further.

g. EVENT TO PASS THROUGH GATES CLEAR POINT

A major portion of the lock module's service look-ahead feature is within the EVENT TO PASS THROUGH GATES CLEAR POINT. This portion of the service look-ahead feature simulates the decision process of an experienced rational lockmaster in deciding whether to recycle the empty chamber as soon as possible, i.e., as soon as the gates are clear, or to leave it open as it is The lockmaster's first consideration is the possibility that a vessel is waiting in the previous throat. If there is a vessel waiting there, the chamber begins recycling immediately, while the current vessel continues exiting to the lock clear point. If the previous throat is empty the lockmaster logic checks the next queue for a vessel which could possible move into the chamber under a SOQA queue discipline. If the next queue contains a vessel, the current vessel continues moving to the clear point, at which time the queued vessel will begin moving into the chamber. If the next queue is empty, the previous queue is checked. If there is a vessel in the previous queue and the service look-ahead feature is used, the next reach is checked for an imminent arrival. If there is no vessel approaching the lock from the next reach, the vessel currently waiting in the previous queue begins moving into a short-entry position while the chamber begins an empty recycle. The same sequence of events occurs if there is an approaching vessel in the next reach which is not expected to be in the chamber before the vessel from the previous queue could. If, however, the next-reach vessel is expected to be in the chamber earlier than the previous-queue vessel, the current vessel simply continues its exit, leaving the chamber open for the imminent arrival. If the previous queue is empty, this means that the lock is idle, or, at least, will be idle as soon as the exiting vessel reaches the clear point If the service look-ahead feature is not used, there is nothing to be done except to complete the current vessel's exit. Otherwise, the previous reach is checked. If there is no imminent arrival in the previous reach, nothing is done. However, if the previous reach does contain an approaching vessel, the next reach is checked. If the next reach is revealed to be empty, the chamber begins recycling to receive the vessel approaching from the previous reach. If the next reach contains an imminent arrival, and it is expected to be in the chamber before the vessel from the previous reach, the chamber remains open as it is to await the arrival. If the previous-reach vessel is expected to be first into the chamber, recycling begins immediately to receive the arrival (see Figure 15).

2 LAKE: a Link Module

The LAKE module, in comparison with the LOCK module, is very simple. It is composed of two routines: the ROUTINE TO SAIL ACROSS LAKE and the EVENT TO EXIT LAKE. The function of the former is to receive a vessel upon arrival at a lake and to schedule the vessel's exit from the lake, while the latter serves to remove a vessel from a lake and to summon the MOVEMENT CONTROLLER to decide the vessel's next move. Associated with each lake in a simulated network is an array of internodal distances which gives the distance between any two nodes located on a lake.

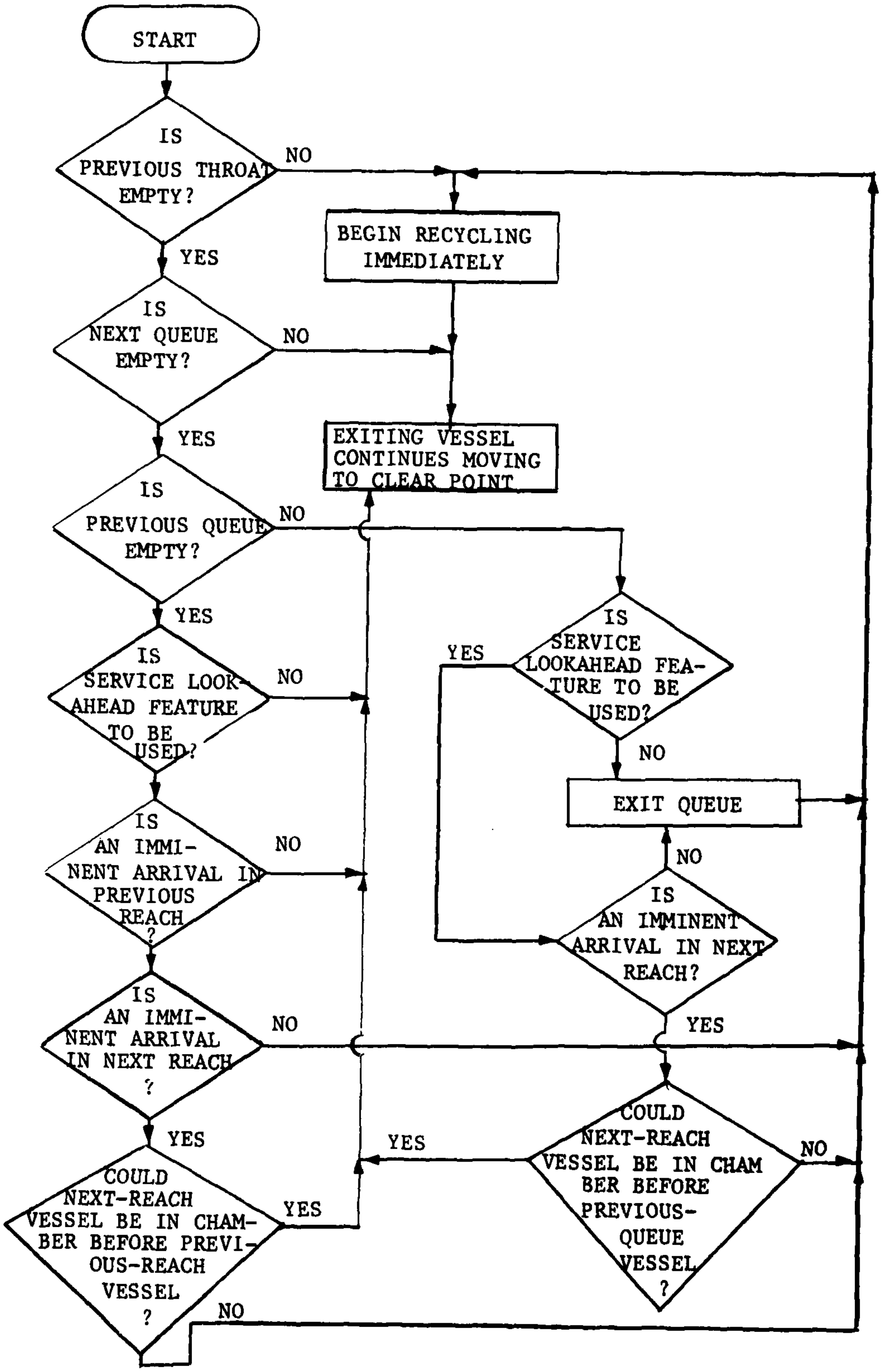

START
IS PREVIOUS THROAT EMPTY?
NO
YES
BEGIN RECYCLING IMMEDIATELY
IS NEXT QUEUE EMPTY?
NO
YES
EXITING VESSEL CONTINUES MOVING TO CLEAR POINT
IS PREVIOUS QUEUE EMPTY?
NO
YES
IS SERVICE LOOK-AHEAD FEATURE TO BE USED?
NO
IS SERVICE LOOKAHEAD FEA-TURE TO BE USED?
YES
NO
EXIT QUEUE
IS AN IMMI-NENT ARRIVAL IN PREVIOUS REACH ?
NO
IS AN IMMINENT ARRIVAL IN NEXT REACH?
NO
YES
IS AN IMMI-NENT ARRIVAL IN NEXT REACH ?
NO
YES
COULD NEXT-REACH VESSEL BE IN CHAM BER BEFORE PREVIOUS-QUEUE VESSEL ?
YES
NO
COULD NEXT-REACH VESSEL BE IN CHAM-BER BEFORE PREVI-OUS-REACH VESSEL ?
YES
NO

Figure 15 General Logic Flow of the EVENT TO PASS THROUGH GATES CLEAR POINT

3. REACH: a Link Module

The REACH module also consists of two routines. The ROUTINE TO SAIL THROUGH REACH schedules a vessel's exit from a reach. This scheduling takes into account the a priori condition regarding passing in the reach. The EVENT TO LEAVE REACH simply removes the vessel from a reach and invokes the MOVEMENT CONTROLLER to assign the vessel's next link.

4. PORT: a Node Module

The port module is unique in two respects. First, it is the only entity in NETSIM/SHIP which can originate and terminate vessels. Thus it has the dual role of both being a node and a facility. Second, the port module as currently implemented in the model is extremely primitive in that it does not to any great extent simulate the complex operations of a port. The skeleton frame for such a simulator however, is present in the port module.

Two routines, the ROUTINE TO ENTER PORT and the EVENT FOR PORT EXIT comprise NETSIM/SHIP's port module. The former receives a vessel at the port and subsequently, either schedules its departure at a randomly determined time in the future, or destroys it. The latter option is included for the following reasons. First, it may be desirable to perform a simulation with a fixed number of vessel trips where the trips are generated by an external program such as TOWGEN. In this case, the external program acts as the scheduler, i.e., the external program determines the schedules of the fleet in advance and inputs them to NETSIM/SHIP so that each vessel is destroyed upon the termination of its journey. Second, it may be desirable to study the effects of vessel interarrival patterns at a particular facility such as a lock. In this case, the interarrival pattern would be determined in advance and vessels would need to be destroyed upon trip termination so as not to interfere with the a priori interarrival pattern. The former option, that is, the ability of a port to schedule departures, provides a framework for a future highly elaborate scheduling mechanism. This consideration will be discussed in the next chapter.

The PORT EXIT routine removes the vessel from the port and transfers the vessel's control to the TRAFFIC CONTROL module. The routine contains two features which render it useful for future expansion. First, it allows the simulation of a scheduling mechanism based on random access of ports. That is, if a vessel has no predefined itinerary, its next port is determined randomly from a user supplied frequency distribution of trips to every other port in the system. Thus, in the case of a two port system, the probability of a trip from one port to the other is exactly equal to 1 since the origin from one port implies a destination at the other. If there are more than two ports, however, the number of possible destinations for any given origin port is greater than one, hence, a frequency distribution of trips would be needed. The random access schedule is, of course, primitive, and can be replaced by a more sophisticated mechanism. On the other hand, if a vessel has a predefined itinerary, the random access scheduler is bypassed and the vessel's next port of call is selected from its itinerary.

The PORT EXIT routine also allows the simulation of season closing. Before scheduling a vessel for another port, the PORT EXIT routine determines whether the simulated shipping season has ended. If it has, the vessel, instead of being rescheduled, is placed into winter berth at its current port The simulation ends when all vessels have entered winter berth.

5. VESSEL: A System Entity

The vessel is the reference point for all the events during a NETSIM/SHIP simulation. It is the movement of vessels, rather than the operations of the network facilities, which is recorded during a simulation. This section describes the attributes of a NETSIM/SHIP vessel and the sets in which a vessel may be filed as it transits network links.

The entity vessel has an attribute, ID VESSEL, which is its identification number. The only restrictions on the vessel entity are that the ID VESSEL be no more than four digits and that each vessel have a unique identification number (this establishes a maximum fleet size of 9999. The four-digit requirement can be altered, if necessary, by changing the formats of NETSIM/SHIP's write statements (refer to the Appendix). VESSEL's second attribute is one giving its OVERALL LENGTH. This attribute is used in NETSIM/SHIP's ALTERNATIVE SELECTOR routine to ascertain that each lock in an alternative route is able to process a vessel of that length. It is also used to affect a vessel's link transit time. VESSEL's CARGO TYPE attribute describes the cargo carried by a vessel and can be used by an expanded scheduler logic to determine a vessel's next port by matching the vessel's cargo with a port's cargo-handling capabilities. This is discussed more fully in Chapter IV under Future Extensions. VESSEL's next four attributes, PREVIOUS NODE, CURRENT NODE, NEXT NODE, and DESTINATION, comprise a vessel's map attributes. These are used, in conjunction with network map information, by the traffic control module in determining a vessel's current location and the next network facility through which the vessel must transit in order to reach its destination. A vessel's PREVIOUS NODE, CURRENT NODE, and NEXT NODE change as the vessel moves from link to link through a network, while a vessel receives a new DESTINATION only when it has entered, and is about to exit, its current DESTINATION. Giving a vessel its new DESTINATION is the job of NETSIM/SHIP's rescheduling logic VESSEL's next two attributes, SELECTED BRANCH and NODES LEFT IN BRANCH, are closely related to a vessel's map attributes. SELECTED BRANCH is used by the ALTERNATIVE SELECTOR to determine which, if any, branch of a set of parallel routes a vessel is transiting. NODES LEFT IN BRANCH tells the ALTERNATIVE SELECTOR where a vessel is located within a route which is parallel to other routes. Finally, VESSEL has an attribute, TIM ARRIVAL, which gives the time at which a vessel arrives at a particular link This is used, along with the time at which a vessel exits from the link, to determine the time required by the vessel to transit the link.

As a vessel moves through a network, it is filed into various sets which represent the different links being simulated. For example, a vessel moving through a lock might first enter a queue, then move through a throat into a chamber, then through an exit throat past another queue and out of the lock. These lock sections are represented in NETSIM/SHIP by the following five sets, respectively: 1 QUEUE, 1 THROAT, CHAMBER, 2 THROAT, and 2 QUEUE. The lock transit example can now be explained as a series of set filings and removals. An arriving vessel might be filed into the 1 QUEUE set of a lock until traffic conditions at the lock are appropriate for the vessel to begin being processed through the lock. At such time, the vessel is removed from the 1 QUEUE set and filed into the 1 THROAT set while the time required to transit the entrance throat elapses. At the end of the transit throat time period, the vessel is removed from the 1 THROAT set and filed into the CHAMBER set of the lock. After the time necessary for a lock's chamber to cycle, the vessel is removed from the CHAMBER set and filed into the 2 THROAT set, the equivalent of the vessel's exit throat. At the end of this exit throat transit time, the vessel passes the lock's second queue, represented by the 2 QUEUE set, which may contain vessels waiting to transit the lock in the opposite direction. These sets, and their relationships to other NETSIM/SHIP sets, are illustrated in Figure 16.

A vessel's transit through NETSIM/SHIP's other links, the REACH and LAKE, as well as the PORT, are also represented by a series of set filings and removals. The NETSIM/SHIP REACH is composed of two sets, 1 DIRECTION and 2 DIRECTION, representing the two directions of travel through a reach. As a vessel enters a reach, it is filed into either the 1 DIRECTION set or the 2 DIRECTION set, depending upon its direction of transit. NETSIM/SHIP's LAKE simulator consists of but one set, ON LAKE TRAFFIC, since direction of transit is not considered to affect lake transit time. There are two sets, DOCK and WINTER BERTH, associated with the PORT model. DOCK is the set into which a vessel is filed during the time period simulating a port stay. WINTER BERTH is the set which recives those vessels that are in port at the end of a simulated season

D. Support Routines

The support module came about as a result of an attempt to reduce duplication within the NETSIM/SHIP logic structure. There are several areas within NETSIM/SHIP where identical, or similar, logic is used to perform certain repetitious tasks Each task is therefore assigned a routine, and and the support module is composed of five such routines.

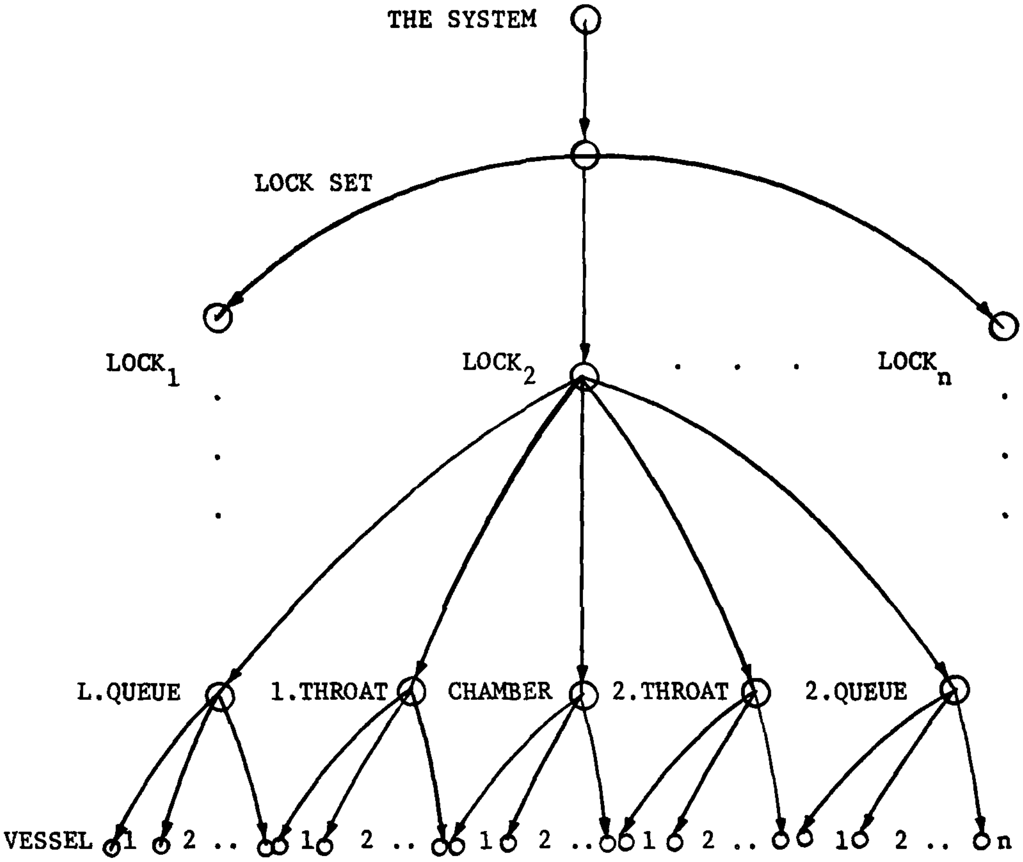


Figure 16. NETSIM/SHIP Set and Entity Relationships

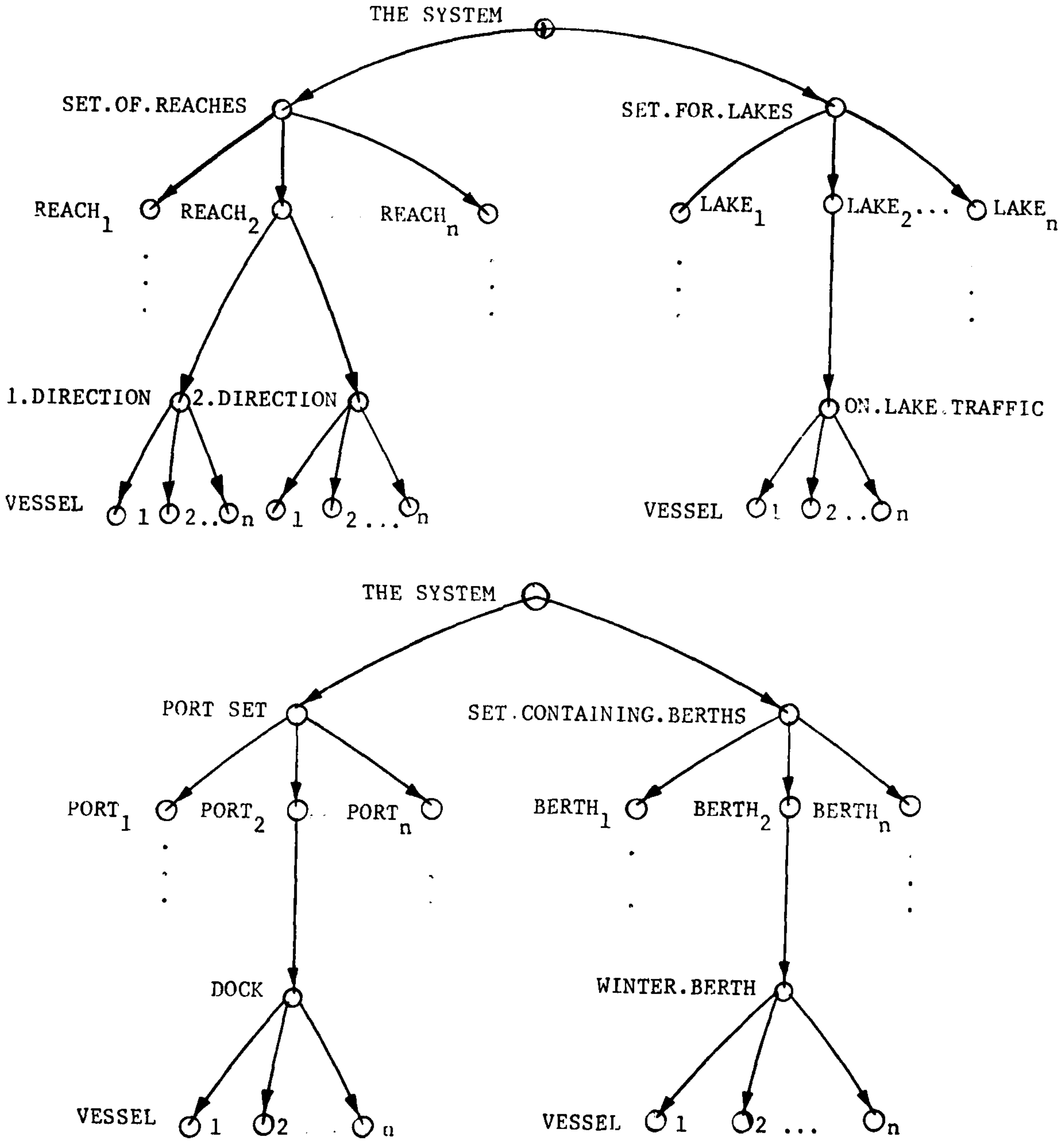


Figure 16. (Continued)

1. ROUTINE TO SEARCH TABLE OF NEXT NODES

Called by the MOVEMENT CONTROLLER routine, the ROUTINE TO SEARCH TABLE OF NEXT NODES receives, as arguments from the MOVEMENT CONTROLLER, a vessel's CURRENT NODE and its DESTINATION. The routine searches down the vertical axis of the TABLE OF NEXT NODES until it locates a node identification number which matches that of the vessel's CURRENT NODE. It then searches across the horizontal axis until it finds a match for the vessel's DESTINATION. The indices identifying the row and column, respectively, of the TABLE OF NEXT NODES where the matches occur are returned to the MOVEMENT CONTROLLER as arguments.

2. ROUTINE TO REFERENCE FACILITIES TABLE

The MOVEMENT CONTROLLER calls upon the ROUTINE TO REFERENCE FACILITIES TABLE to determine the identification number for a vessel's next facility. The routine first searches down the vertical axis of the FACILITIES ID TABLE until it finds a node identification number which matches that of the vessel's CURRENT NODE. The horizontal axis is then searched for a match to the vessel's NEXT NODE. The indices identified by these matches locate, within the body of the table, the identification number of the link which connects the vessel's CURRENT NODE with its NEXT NODE. This identification number is returned to the MOVEMENT CONTROLLER as the vessel's next facility.

3. ROUTINE FOR STOCHASTIC TIME CALCULATIONS

During simulation, whenever a random time is needed for any of the NETSIM/SHIP time blocks, the ROUTINE FOR STOCHASTIC TIME CALCULATIONS is called to provide the random time element. Altogether, NETSIM/SHIP includes twelve different time blocks, which requires that the ROUTINE FOR STOCHASTIC

TIME CALCULATIONS be given a code by the calling routine to identify the proper time block to be used. The routine, after identifying the proper time block, determines whether the distribution function has been provided empirically by the user or whether it is to be one of the eleven SIMSCRIPT-supplied theoretical distributions. If the distribution is an empirical one, the routine samples from it using SIMSCRIPT's random variable mechanisms. On the other hand, if the distribution is a SIMSCRIPT theoretical function, a separate section of the ROUTINE FOR STOCHASTIC TIME CALCULATIONS is called to perform the sampling. Finally, the adjusted time value is returned by the ROUTINE FOR STOCHASTIC TIME CALCULATIONS to the calling routine

There are eleven statistical distribution functions supplied by the SIMSCRIPT system (see Kiviat, page 314). Each of these eleven functions is available to a NETSIM/SHIP user for describing any of the time blocks within the model (refer to the Appendix). This availability is made possible through the logic to calculate theoretical random time which calls a particular SIMSCRIPT function and returns to the calling routine the random value generated by the function

4. ROUTINE TO USE ATTRIBUTES TO ADJUST TIME

Designed to permit link transit times to reflect the effects of attributes of the system entities involved, the ROUTINE TO USE ATTRIBUTES TO ADJUST TIME is, in the present implementation, a demonstration of the potential for, rather than an actual, use of the time-affecting attributes.

5. ROUTINE TO QUERY INTRALAKE DISTANCES TABLE

The final routine in the support module is used by the lake module and by the ALTERNATIVE SELECTOR to include the effects of distance in the computation of a lake transit time. The routine searches through the INTRALAKE

DISTANCES TABLE (refer to Figure 17) until it locates the proper lake, current node, and next node. These three indices provide the distance from the current node to the next node. This distance is returned to the calling routine.

LAKE ID	NODE ID_1	NODE ID_2	NODE ID_3	. . .	NODE ID_m**
NODE ID_1	0	IND^*_{12}	IND_{13}	. . .	IND_{1m}
NODE ID_2	IND_{21}	0	IND_{23}	. . .	IND_{2m}
NODE ID_3	IND_{31}	IND_{32}	0	. . .	IND_{3m}
⋮	⋮	⋮	⋮	⋮	⋮
NODE ID_n**	IND_{n1}	IND_{n2}	IND_{n3}	. . .	0

*Inter-Nodal Distance (IND)
**m = n

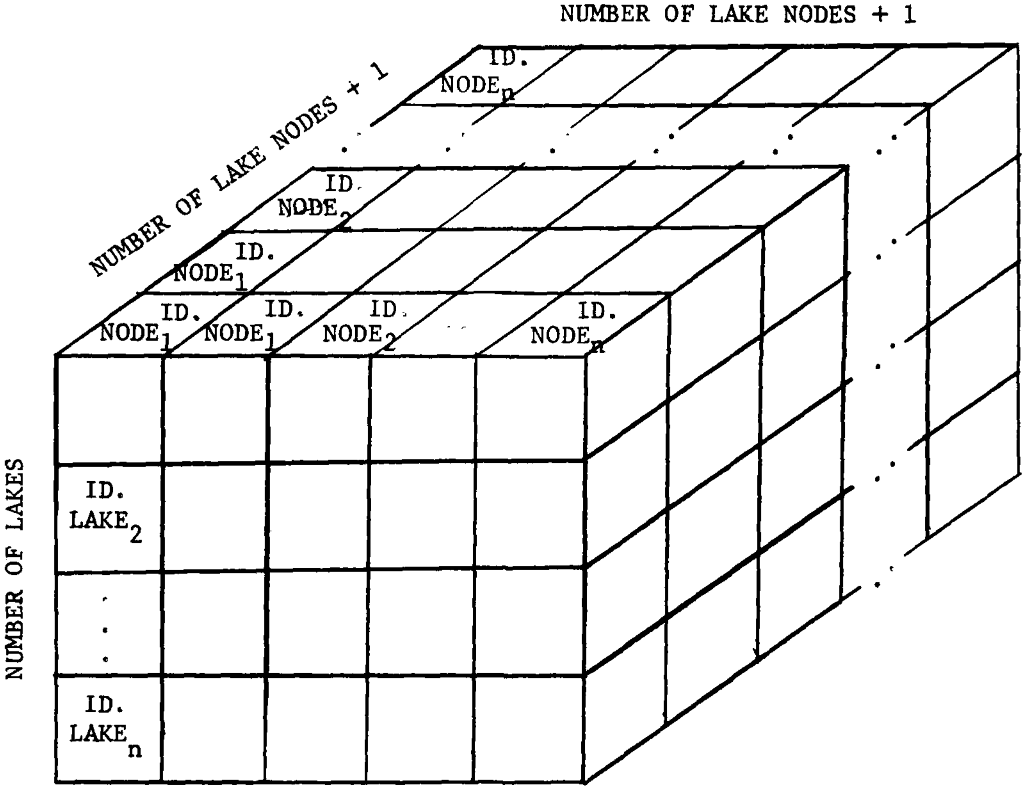


Figure 17. INTRALAKE.DISTANCES.TABLE: Array Storage for Lakes' Internodal Distances

IV. RESEARCH POSSIBILITIES AND FUTURE EXTENSIONS

The increasing levels of traffic experienced by the Great Lakes-St. Lawrence River Seaway have brought to forefront the need for a careful assessment of the waterway's ability to meet the demands placed upon them within the foreseeable future. Because of the extremely high cost and long life of structural improvements, the need for accurate and detailed research into structural, as well as non-structural, alternative improvements is easily recognizable. The current implementation of NETSIM/SHIP, as described in this report, provides a powerful tool for use in conducting the needed research.

The potential applications for NETSIM/SHIP involve primarily investigations into the impact of both structural and non-structural alternative improvements. NETSIM/SHIP permits the user to consider the effects upon the system of various fleet characteristics, varying traffic levels, new and/or improved facilities, changing commodity flows, different facility locations, ship scheduling procedures, dock strikes, and changes in the length of the shipping season

To conduct research into some of the areas mentioned requires more sophistication in the model than is currently available in NETSIM/SHIP. However, the development of NETSIM/SHIP is a continuous undertaking and, at the time of this writing, efforts are being made to extend the model into the proper levels to enable a full, system-wide investigation of the problem areas. The extensions to be implemented during the earlier developmental phases include a port model which determines a vessel's turn-around time as a function of the characteristics of the vessel and of the port, a rescheduler model which selects a vessel's next port as a function of the vessel's cargo-carrying capabilities and the ports' cargo-availability characteristics, and a post-simulation processor to prepare NETSIM/SHIP's two output data bases for statistical analysis. Later phases of NETSIM/SHIP's development could include an elaborated post-simulation processor to produce graphical displays of simulation results, an expanded reach model to explicitly take into consideration the effects of reach width, depth, and sinuosity, an additional lock model for explicitly simulating the operations of shallow-draft locks, and a capability for a user to conveniently select, through input parameters, one of several specific facility models.

With the implementation of the suggested extensions, NETSIM/SHIP would be capable of supporting research with a system-wide perspective of the effects of improved port facilities. In its current implementation, the model can be used for studying alternative lock designs, as well as alternative canal designs for developing parallel transit routes. Studies of lock capacities under varying conditions of traffic and fleet composition can be conducted. An investigation of the implications of an extended length for the shipping season could be carried out using the improved NETSIM/SHIP.

V. SUMMARY

The model described in this report is an extension of a navigation systems simulator developed by the Pennsylvania Transportation and Traffic Safety Center. The initial model (hereafter referred to as MCDD) is described by Rea and Nowading in their research report, Simulation of Multiple Channel Deep Draft Navigation Systems, (1971). The authors describe the features which, if implemented, would greatly increase the usefulness of the model as a research tool.

The unique feature of the MCDD is its capability for simulating concurrent, two-way traffic movements through multiple parallel canals. This capability is made possible by the use of a unique assignment map technique. The assignment map feature involves the generation of an experience data base which is statistically analyzed to derive expected transit time functions. Expected transit times are computed for each alternative canal based on current conditions within the canal. A vessel is assigned to the canal yielding the least expected transit time.

The MCDD model includes the capability for simulating locks and reaches. Its lock simulator uses eight probability distributions to simulate time lapses during a lock's operations Each lock is allowed to have a distinct set of distributions. Queues are served alternately (a SOQA queue discipline) until there exists but one queue, at which time the queue discipline is FCFS. The model includes a recycle "look-ahead" feature, which recycles an empty chamber, given certain conditions, to minimize a vessel's delay. The model's reach simulator can use one of two distributions for a reach's transit time to take into consideration any directional differences in transit times.

A no-passing rule is maintained in all reaches, thus allowing an entering vessel to catch, or fall farther behind, an immediately preceding vessel, but not to get ahead of it

The logic structures providing the MCDD model's capabilities are conceptually modularized. The modular framework groups closely related logic structures, thereby permitting any particular module to be altered with a minimal amount of disturbance to the other modules. The model is programmed using GPSS (General Purpose Simulation System), an easily-learned, problem-oriented, discrete-event, simulation language.

The basic justification for the current project is the desire for a generalized transportation model. The concepts developed during the MCDD model's evolution can be used to describe any network system comprised of nodes and links. However, because of the inefficiency and extreme physical size that would result if GPSS were used, the MCDD model cannot be conveniently expanded to simulate any arbitrary transportation network. Because it is a general-purpose, multiple-level (upper levels are designed especially for discrete-event simulation applications) programming language, SIMSCRIPT is used in developing the extended, generalized model (hereafter referred to by the acronym NETSIM/SHIP, NETwork SIMulator applied to SHIP movements).

NETSIM/SHIP embodies all the features of the MCDD model described above, as well as many modifications and extensions. NETSIM/SHIP differentiates between directions of travel by sampling from distinct distributions, each of which can be empirical or theoretical according to the user's selection. All random times can be modified to include effects of the system entities involved. The lock model contains an "experienced lockmaster" simulator which, when no queues exist, inspects adjacent reaches for imminent arrivals and commits the lock to that vessel with the earliest expected arrival time, recycling the chamber if necessary. This ability is referred to as the service look-ahead feature. A second ability of the "experienced lockmaster" simulator is referred to as the recycle look-ahead feature. Whenever a ship arrives at an idle lock, NETSIM/SHIP assumes that the "experienced lockmaster" would have seen the arriving ship and would have committed the lock to it, recycling if necessary, in order to minimize the ship's delay. The reach simulator features an "on-off" switch for the passing rule. With the passing rule switched "on," an arriving ship can get ahead of preceding ships in the reach. Each reach can have the passing rule on or off, as the user desires. NETSIM/SHIP also features a realistic shipping season simulation model. Each ship can begin the shipping season from a distinct port at an individually determined time. At season's end each ship is placed in winter berth at a port.

NETSIM/SHIP extends the MCDD model by providing a lake simulator and a port simulator. The lake simulator is somewhat similar to the reach simulator in that a ship enters at one node, receives a random transit time, and exists at another node. The lake transit times are determined using an array of intralake internodal distances and attributes of the lake and ship. Direction of transit has no effect. Ship passings are always allowed. The port simulator is included in NETSIM/SHIP primarily as a demonstration of the potential of such a model. A realistic simulation model of port activities requires considerable research and programming. The current port simulator does, however, provide the basic mechanisms for a more realistic model. The port accepts a ship and determines a random time for its stay in port. The port simulator includes a "season-ending" feature which is invoked prior to a ship's exit from a port. If, when a ship is about to exit a port, the shipping season has ended, the ship is placed into winter berth at that port. If the season has not ended, the ship's new destination port is obtained from the NETSIM/SHIP "scheduler" feature. Each ship, when it enters the system, can have a pre-assigned itinerary. The scheduler references the array of ship itineraries and, if the exiting ship is listed therein, assigns the ship to the port indicated on its itinerary. If a ship has no preassigned itinerary, the scheduler randomly selects the ship's next port using a probability distribution of available ports. The scheduler feature can be expanded to include attributes of ports and the ship in selecting a ship's next port.

The logic structures of NETSIM/SHIP, like those of the MCDD model, are conceptually modularized, i.e., structures of similar function are grouped together. The complete logical framework is segregated into the following four modules:

1. Initialization
2. Traffic control
3. Support
4. Link.

The logic flow of the NETSIM/SHIP process can be described in terms of interactions among the conceptual modules.

Prior to the beginning of simulated time, the initialization module reads in parameters and system entity attributes, sets up the simulated system and schedules each ship to exit its designated port at its designated time. The port module, upon each ship exit, calls the traffic control module Using NETSIM/SHIP's dynamic course determination feature, the traffic control module determines each ship's next link (lock, lake, or reach) and calls the appropriate logic module. When a ship reaches its next node, the traffic control module is called again. The node reached by the ship may be its destination port and, if it is so determined by the traffic control module, the port module is called upon to receive the ship; otherwise, the traffic control module determines the ship's next link and calls the appropriate module. This cycle of interactions among the modules continues until the end of the simulated shipping season. The support module is called throughout the simulation by the other modules for assistance in array searching and random time generation.

The input data required by NETSIM/SHIP are divided into four categories: (1) parameters needed for a particular run, (2) parameters giving the dimensions (numbers of lakes, locks, vessels, etc.) of the simulated system, (3) attributes of the system entities, and (4) arrays describing the layout of the system. The output data supplied by NETSIM/SHIP are of basically two categories: (1) the Experience Data Base (EDB), and (2) the EVENT LOG. The EDB contains prior canal conditions for each ship arrival and subsequent ship transit times. The EDB is statistically analyzed to derive functions used to compute expected transit times during a simulation. These expected transit time functions are the basis for NETSIM/SHIP's capability for dynamically determining a ship's course through multiple parallel canals. The EVENT LOG is generated during a simulation run, giving data that describe each eventoccurrence during the run. The LOG is statistically analyzed to determine system performance relative to whatever parameters the user is interested in.

In its current state of development, NETSIM/SHIP can be used in numerous research projects involving simulation of deep draft waterways. The detailed lock module permits lock design studies. Studies can be done on the system effects of lock designs and lock location. Alternate canal configurations can be investigated. With the development of a realistically detailed port module, NETSIM/SHIP would be capable of commodity flow studies, studies on the implications of port design and commodity handling equipment configurations, and the effects of ship design on system and port utilization, as well as many other projects involving port system simulation. A refined reach module, which would include consideration for such factors as reach width, depth, number of bends, etc., would increase NETSIM/SHIP's capabilities. The true power of the model, however, lies in the conceptual framework upon which NETSIM/SHIP is built. The conceptualization of the traffic control module is so general that attributes of particular traffic entities, and attributes of particular nodes and links, have no effect on the capabilities of NETSIM as a network simulator. This means that, with the development of the proper simulator modules, NETSIM can be used for research applications involving, for example, personal rapid transit systems, mass urban transit systems, or airport design. As previously stated, if a network system can be described in terms of nodes and links, NETSIM can be used to simulate it.

REFERENCES

1. Carroll, Joseph L., and Bronzini, Michael S., "Waterway Systems Simulation: Volume 1--Summary Report," Vol. I of a series, Report TTSC 7108, Pennsylvania Transportation and Traffic Safety Center, The Pennsylvania State University, University Park, Pennsylvania, 1971.

2. Rea, John C., and Nowading, David C Waterway Systems Simulation: Volume V--Simulation of Multiple Channel Deep Draft Navigation Systems. Report TTSC 7112, Pennsylvania Transportation and Traffic Safety Center, The Pennsylvania State University, University Park, Pennsylvania, 1971.

3. Carroll, Joseph L., "Waterway Lock Simulation Model," Papers, Sixth Annual Meeting, Transportation Research Forum, 1965.

4. Hazard, John L. The Great Lakes-St. Lawrence Transportation System--Problems and Potential. Upper Great Lakes Regional Commission, Washington, D.C., 1969.

5. The St. Lawrence Seaway Authority and St Lawrence Seaway Development Corporation Traffic Report of the St. Lawrence Seaway, 1970. Ottawa, Canada: Queen's Printer, 1970.

6. Bronzini, Michael S. Waterway Systems Simulation: Volume II--TOWGEN: A Tow Generation Model for Inland Waterway Simulation. Report TTSC 7110, Pennsylvania Transportation and Traffic Safety Center, The Pennsylvania State University, University Park, Pennsylvania, 1971.

7. Kiviat, P J., R. Villanueva, and H. M. Markowitz, The SIMSCRIPT II Programming Language, Prentice-Hall Inc , Englewood Cliffs, N.J., 1968.

Appendix

NETSIM/SHIP

USER'S MANUAL

Table of Contents

A. Introduction

NETSIM/SHIP is a general network simulation model capable of simulating concurrent, bi-directional traffic through multiple, parallel routes. The model is programmed in SIMSCRIPT II.5, a general-purpose simulation language. Because of the highly technical nature of NETSIM/SHIP the user is assumed to be familiar with the general areas of research (presently, waterway simulation) in which the model is being applied. In addition, the user should have access to the following two documents:

1. Kiviat, P. J., R. Villanueva, and H. M. Markowitz. The SIMSCRIPT II Programming Language. Prentice-Hall, Inc., Englewood Cliffs, N. J., 1968.

2. Consolidated Analysis Centers, Inc. SIMSCRIPT II.5 User's Manual. Consolidated Analysis Centers, Inc., Los Angeles, Calif., 1971.

The purpose of this User's Manual is to provide a guide by which a user can be insured of properly preparing the necessary input data stream for NETSIM/SHIP and of properly interpreting the output data bases. The manual is divided into basically two sections. The first section describes, in detail, NETSIM/SHIP's required input data stream. NETSIM/SHIP's two output data bases, as well as its error messages, are covered in the second section.

B Input Data Stream

Before describing the actual input data stream, a brief mention of SIMSCRIPT's free-form READ statement (Kiviat, page 7) is in order. There are only two restrictions concerning the format of the input data stream. First, each number in the stream must be separated from preceding and succeeding numbers by at least one blank (there may be any number of blanks above one). Second, a number must not be continued from one card to the next (a number terminates at the end of a card). Numbers may be punched as either real or integer. However, if a real number has a fractional part, its decimal must be included. Since SIMSCRIPT treats input data as a continuous stream of numbers, a number's particular location on a card is not considered, as long as it is properly positioned relative to the other numbers in the stream. The correct relative positions for the input data received for NETSIM/SHIP are described in the remainder of this section.

The data required by NETSIM/SHIP are divided into the following four categories:

1. Run Parameters
2. System Size Parameters
3. System Entity Descriptors
4. System Network Descriptors

Explanation of each category, along with specifications for each datum, are given below.

1. Run Parameters

The run parameters provide NETSIM/SHIP with information pertaining to available options.

Datum	Description
1.1	the length of the "season" being simulated in minutes.
1.2	1, if the run generates an Experience Data Base (EDB) 0, if the run generates a simulation event LOG (LOG).
1.3	1, if the run uses NETSIM/SHIP's service look-ahead feature for dynamic queue selection. 0, if the service look-ahead feature is not used.
1.4	1, if the system being simulated contains one or more facilities with parallel alternative facilities. 0, if the system has no parallel facilities.
1.5	the device number for the output unit upon which NETSIM/SHIP writes its output data bases. It can be any number between 1 and 15, excluding 2, 3, and 5. This number must agree with the number given on the Data Definition (DD) card (for IBM OS/360 JCL).

Datum	Description
1.6	the device number for the input unit from which NETSIM/SHIP receives the necessary external data for generating system arrivals.
1.7	1, if vessels are to enter ports, be re-scheduled, and continue on their routes. 0, if vessels are to be destroyed upon port entry.
1.8	1, if necessary data for scheduling system arrivals are to be supplied exogenously during program execution. 0, if these data are to be supplied with standard input data stream.

2. System Size Parameters

The size parameters determine the extent of the system being simulated, and provide appropriate amounts of memory space within the computer. Each datum must be an integer value, with its maximum determined by the capacity of the equipment (computer) being used.

The system, in NETSIM/SHIP, is defined as a network of links and nodes. Every facility (lock, reach, lake, etc.) is modeled as a link. Every link must be bounded on either side by a node which delineates any two links. A node may either be a port or an arbitrarily defined link delimiter. Only a port node may originate and terminate vessel journeys Non-port nodes, however, act as decision points where

a) the vessel decides whether alternative routes exist.

b) the vessel determines the type of facility that must be negotiated.

Non-port nodes are thus useful in providing the vessel with, in essence, a map of its journey.

The format of the system parameters is as follows:

Datum	Description
2.1	the number of ports
2.2	the number of lakes
2.3	the number of reaches

Datum Description

2.4 the number of locks

2.5 the number of vessels (set equal to 1, if datum 1.8 = 1).

2.6 the number of nodes (includes the number of ports plus the the number of non-port nodes).

2.7 the number of vessels that have pre-assigned itineraries.

3 System Entity Descriptors

The system entity descriptors are composed of data for each port, lake, reach, lock, and vessel included in the simulated system. The data must be given in a specific order, i.e., all ports, then all lakes, then all reaches, then all locks, and then all vessels. Within this arrangement, however, order is not critical (e.g., it does not matter whether the data for port n is read before or after the data for port n-1, as long as both data streams are composed according to the specifications given below, and both streams precede those for vessels).

Datum Description

3.1 the port identification number (any integer from 1 to 9999).

3.2 an integer (1-9999) identifying either the type or amount of commodity held at the port. The datum is not actually used during the simulation run, but is included to demonstrate the possibility of port commodity manipulation.

3.3 a Cumulative Distribution Function (CDF) identifying the probabilities of a vessel's transiting to any other port in the system. Because the datum is defined by NETSIM/SHIP as a Random Step Variable (RSV, refer to Kiviat, page 316), it must be followed by an asterisk (*), e.g., 0.1 1022 0.2 1032 0.35 1042 0.55 1052 0.8 1062 1.0 1072 *.

3.4 1, if the CDF describing the time-lapse for a vessel's stay in port is one of the SIMSCRIPT-supplied theoretical CDF's (see Kiviat, page 314) 0, if the CDF is empirical and is included as part of the input data stream.

Datum Description

3.5 if datum 3.4 = 1, datum 3.5 is composed of the following items:

(1) identification number of the port
(2) an integer (1-11) code referring to the selected SIMSCRIPT-supplied CDF (refer to Table 1 below and Kiviat, page 314)
(3), (4) and (5) arguments for the SIMSCRIPT-supplied CDF as described in Kiviat, page 314. If the CDF specifies only two arguments, item (5) must still be included in the input stream as 0 (zero).

If datum 3.4 = 0, datum 3.5 is the empirical CDF, defined and read by NETSIM/SHIP as an RSV (refer to the discussion on RSV's in 3.3 above)

Data 3.1 through 3.5 are repeated for each port in the simulated system.

3.6 the lake identification number (1-9999).

3.7 the number of nodes (including ports) connected to the lake.

3.8 1, if the lake is a link within a set of parallel links; 0, if not

3.9 if the CDF describing the time-lapse for a vessel's lake transit is one of the SIMSCRIPT-supplied theoretical CDF's, datum 3.9 is 1 (one). If the CDF is empirical and is included as part of the input data stream, then 3.9 is 0 (zero)

3.10 if datum 3.9 is 1, datum 3.10 is composed of the following five items:

(1) identification number of the lake
(2) an integer (1-11) code referring to the selected SIMSCRIPT-supplied CDF (refer to Table 1 above)
(3), (4), and (5) arguments for the SIMCRIPT-supplied CDF as described in Kiviat, page 314. If the CDF specifies only two arguments, item (5) must be included in the input stream as 0 (zero).

If datum 3.9 is 0, datum 3.10 is the empirical CDF, defined and read by NEISIM/SHIP as a RSV (refer to the discussion on RSV's in 3.3 above).

3.11 Datum 3.11 is optional, dependent upon, first, whether the run is EDB or LOG and, second, whether the lake exists as a parallel link. If Datum 1.2 is 0 (zero) and Datum 3.8 is 1 (one), Datum 3.11 consists of the following 9 items.

TABLE 1

NETSIM/SHIP Codes for
Statistical Distribution Functions

Function	NETSIM/SHIP Code
BETA. F	1
BINOMIAL. F	2
ERLANG. F	3
EXPONENTIAL. F	4
GAMMA. F	5
LOGNORMAL. F	6
NORMAL. F	7
POISSON. F	8
RANDI. F	9
UNIFORM. F	10
WEIBULL.F	11

Datum Description

(1) identification number of the lake
(2) a coefficient to be used to weight the effect upon expected transit time of the number of vessels on the lake when the vessel makes its decision as to which alternative set of links to use
(3) a coefficient used to weight the effect upon expected transit time of the distance from the vessel's entry node to its exit node
(4) - (9) 0 (these provide the capability for additional factors to be included when, and if, needed).

Any of the factors can be deleted by setting its coefficient equal to 0 (zero).

3.12 Datum 3.12 consists of an Origin-Destination (O-D) table which gives the distance between any two nodes on the lake. The table is read row by row. The first row contains, in its first column, the identification number of the lake. Columns 2 through n of the first row contain the identification of each node (including ports) on the lake. There is no required order for the node identification numbers. The first column of rows 2 through n contains the identification number of each of the n nodes on the lake. Columns 2 through n of rows 2 through n contain individual internodal distances.

Data 3.6 through 3.12 are repeated for each lake in the simulated system.

3.13 the reach identification number (1-9999).

3.14 the length of the reach, in feet.

3.15 1, if vessel passings are allowed within the reach,
0, if passings are not allowed.

3.16 1, if the reach is a link within a set of parallel links;
0, if not.

3.17 1, if the CDF describing the time-lapse for a vessel's reach transit is one of the SIMSCRIPT-supplied theoretical CDF's.
0, if the CDF is empirical and is included as part of the input data stream.

3.18 the identification number (1-9999) of the reach's upstream end-point node.

3.19 the identification number (1-9999) of the reach's downstream end-point node.

Datum Description

3.20 If datum 3.17 is 1, datum 3.20 is composed of the following six items:

(1) identification number of the reach
(2) identification number of the upstream end-point node toward which a vessel sails when its transit time is described by the CDF
(3) an integer (1-11) referring to the selected SIMSCRIPT-supplied CDF (refer to Table 1 above)
(4), (5), and (6) arguments for the SIMSCRIPT-supplied CDF as described in Kiviat, page 314. If the CDF specifies only two arguments, item (6) must be included in the data input stream as 0 (zero). Items (1) through (6) are repeated for the reach's downstream end-point node. The two sets of six items combine to provide a unique CDF for each direction of traffic flow.

If datum 3.17 is 0, datum 3.20 consists of two empirical CDF's, one for each direction of traffic flow in the reach. The first empirical CDF corresponds to 3.18, i.e., 3.18 is the identification number of the reach's end-point node toward which vessels sail when the first CDF is used to describe their transit times. The second CDF must correspond to 3.19. Because each CDF is read by NETSIM/SHIP as a random step variable, each one is followed by an asterisk (*).

3.21 If datum 1.2 is 0 (zero), and datum 3.16 is 1 (one), datum 3.21 is composed of twenty items, ten for each of the two traffic flow directions. The items are entered into the input data stream as follows:

(1) identification number of the reach
(2) identification number of the upstream end-point node toward which the vessel is sailing when its Expected Transit Time (ETT) is determined by the coefficients given in items (3) through (10) below
(3) coefficients used to weight the effect upon expected transit time of the number of vessels sailing in the opposite direction within the reach
(4) coefficient used to weight the effect upon expected transit time of the number of vessels sailing in the same direction within the reach
(5) coefficient used to weight the effect upon expected transit time of the square of the number of vessels sailing in the opposite direction within the reach
(6) coefficient used to weight the effect upon expected transit time of the square of the number of vessels sailing in the same direction within the reach

Datum Description

(7) coefficient used to weight the effect upon expected transit time of the log[1] of the number of vessels sailing in the opposite direction within the reach.

(8) coefficient used to weight the effect upon expected transit time of the log of the number of vessels sailing in the same direction within the reach.

(9) coefficient used to weight the effect upon expected transit time of the square root of the number of vessels sailing in the opposite direction within the reach

(10) coefficient used to weight the effect upon expected transit time of the square root of the number of vessels sailing in the same direction within the reach

Items (1) - (10) are repeated for the downstream direction.

If datum 1.2 is 1 (one) and/or datum 3.16 is 0 (zero), datum 3.21 is not included in the input data stream.

Data 3.13 through 3.21 are repeated for each reach in the simulated system.

3.22 identification number of the lock.

3.23 the identification number (1-9999) of the lock's upstream end-point node.

3.24 the identification number of the lock's downstream end-point node.

3.25 the maximum vessel length, in feet, that can be processed by the lock.

3.26 a code representing whether the lock is within a set of parallel links. The code is 1 if the lock is in a set of parallels, 0 if it is not.

Each lock has associated with it nine different CDF's for determining the random time lapses of different events. Data 3.27 through 3.35 determine whether each of the nine CDF's is a theoretical or an empirical CDF. If the particular CDF is a SIMSCRIPT-supplied theoretical CDF, its datum is 1 (one). If the CDF is empirical and, therefore, supplied with the input

[1] To avoid the problem which arises in the computation of the log of zero, NETSIM/SHIP actually adds 0.001 to the argument before the logarithmic operation.

data stream, its datum is 0 (zero). Each description given below describes the particular time block connected with each datum. The datum is either 1 (one) or 0 (zero), as described above.

3.27 time from clear point to tie-up in chamber, from a moving start.

3.28 time from clear point to tie-up in chamber, from a stationary start.

3.29 time from just outside chamber gates to tie-up in chamber.

3.30 time from tie-up in chamber to point where vessel's stern is clear of chamber's gates.

3.31 time from point where vessel's stern is clear of chamber's gates to clear point.

3.32 time from clear point to position just outside chamber gates, from moving start.

3.33 time from tie-up in queue to position just outside chamber gates.

3.34 time required by chamber for processing.

Data 3.36 through 3.44 below are composed of either (1) arguments for a SIMSCRIPT-supplied theoretical CDF, or (2) the values for an empirically-determined CDF. Each datum consists of either arguments or empirical values depending upon whether its corresponding datum in 3.27 through 3.35 is 1 (one) or 0 (zero), respectively.

3.36 if 3.27 is 1, 3.36 consists of the following six items:

(1) identification number of the lock
(2) identification number of the lock's upstream end-point node toward which the vessel is traveling when its transit time is described by this CDF
(3) an integer (1-11) referring to the selected SIMSCRIPT-supplied CDF (refer to Table 1 above)
(4), (5), and (6) arguments for the SIMSCRIPT-supplied CDF as described in Kiviat, page 314. If the CDF specifies only two arguments, item (6) must be included in the data input stream as 0 (zero).

Items (1) through (6) are repeated for the lock's downstream end-point node.

Datum Description

If 3.27 is 0, 3.36 consists of two empirical CDF's, one for each direction of traffic flow in the lock. The first empirical CDF corresponds to 3.24, i.e., 3.24 is the identification number of the lock's end-point node toward which vessels sail when the first CDF is used to describe their transit times. The second CDF must correspond to 3.25. Because each CDF is read by NETSIM/SHIP as a random step variable, each one is followed by an asterisk (*)

3.37 same as 3.36 except that 3.28 replaces 3.27

3.38 same as 3.36 except that 3.29 replaces 3.27.

3.39 same as 3.36 except that 3.30 replaces 3.27.

3.40 same as 3.36 except that 3.31 replaces 3.27.

3.41 same as 3.36 except that 3.32 replaces 3.27.

3.42 same as 3.36 except that 3.33 replaces 3.27.

Datum 3.43 (for chamber recycle time), as well as 3.44 (for chamber process time), differ from data 3.36 through 3.42 in that their times are not affected by direction of transit. Therefore, only one empirical CDF, or one set of arguments, is given for each datum

Datum Description

3.43 If 3.34 is 1, 3.43 consists of the following six items:
(1) identification number of the lock
(2) 1 (a code for the chamber recycle CDF)
(3) an integer (1-11) referring to the selected SIMSCRIPT-supplied CDF (refer to Table 1 above)
(4), (5), and (6) arguments for the selected SIMSCRIPT-supplied CDF as described in Kiviat, page 314. If the CDF specified only two arguments, item (6) must still be included in the data input stream as 0 (zero).
If 3.34 is 0, 3.43 consists of the empirical CDF, followed by an asterisk (*)

3.44 If 3.35 is 1, 3.44 consists of the following six items:
(1) identification number of the lock
(2) 2 (a code for the chamber process CDF)
(3) an integer (1-11) referring to the selected SIMSCRIPT-supplied CDF (refer to Table 1 above)
(4), (5), and (6) arguments for the selected SIMSCRIPT-supplied CDF as described in Kiviat, page 314. If the CDF specified only two arguments, item (6) must still be included in the data input stream as 0 (zero).
If 3.35 is 0, 3.44 consists of the empirical CDF, followed by an asterisk (*)

Datum Description

3.45 If datum 1.2 is 0 (zero) and datum 3.26 is 1 (one), datum 3.45 is composed of twenty items, ten for each of the two traffic flow directions. The items are entered into the input data stream as follows:

(1) identification number of the lock
(2) identification number of the upstream end-point node toward which the vessel is sailing when its Expected Transit Time (ETT) is determined by the coefficient given in items (3) through (10) below
(3) coefficient used to weight the effects of the size of the near queue upon transit time through the lock
(4) coefficient used to weight the effects of the near throat's status upon lock transit time. Throat status can be one of the following:
(a) 0, if throat is empty
(b) 1, if throat contains a vessel moving in opposite direction
(c) 2, if throat contains a vessel moving in same direction
(5) coefficient used to weight the effects of the chamber's status upon lock transit time. Lock status is either 0, 1, or 2 (chamber is either empty, has a vessel moving in opposite direction, or has a vessel moving in same direction, respectively)
(6) coefficient used to weight the effects of the far throat's status upon lock transit time. Throat status is either 0, 1, or 2 (refer to item 5 above)
(7) coefficient used to weight the effects of the far queue's size upon lock transit time
(8) 0
(9) 0
(10) 0

Items 1 through 10 are repeated for the lock's downstream end-point node.

If datum 1.2 is not 0 (zero) and datum 3.26 is not 1 (one), datum 3.45 is not included in the input data stream.

Data 3.23 through 3.45 are repeated for each lock in the simulated system.

If datum 1.8 is 1 (one), data 3.46 through 3.53 are omitted from the input data stream and data for the fleet of vessels are supplied, during execution externally from the input device given in datum 1.6. Otherwise, if datum 1.8 is 0 (zero), the fleet data consists of the following:

Datum Description

3.46 identification number of the vessel (1-9999)

3.47 overall length of the vessel, in feet.

Datum Description

3.48 an integer code (1-9999) representing either the type or quantity of commodity being carried by the vessel.

3.49 identification of the port from which the vessel is to depart at the beginning of the simulated season.

3.50 0, if the vessel has no pre-determined itinerary of ports to visit;
1, if the vessel has one or more pre-determined ports to visit.

3.51 the time (in minutes) after the beginning of the simulated season when the vessel leaves the port given in 3.49.

3.52 If 3.50 is 1 (one), 3.52 is the number of ports that are pre-determined for the vessel to visit. If 3.50 is 0, 3.52 is not included in the input data stream.

3.53 If 3.50 is 0 (zero), 3.53 is omitted.
If 3.50 is 1 (one), 3.53 consists of the following:

(1) identification number of the vessel
(2) identification number of the first pre-determined port to visit
(3) identification number of the second pre-determined port to visit

. .
. .
. .

(n+1) identification number of the nth pre-determined port to visit (n is given in 3.52).

Data 3.46 through 3.53 are repeated for each vessel in the simulated system.

4. Network Configuration Descriptors

The fourth and final section of the input data stream is composed of the following four tables:

(1) TABLE OF NEXT NODES

(2) FACILITIES ID TABLE

(3) PARALLEL FACILITIES TABLE

(4) ALTERNATIVE WIDE COEFFICIENTS

They are entered into the input data stream as described below:

Datum — Description

4.1 TABLE OF NEXT NODES, section 1 Section 1 is an Origin-Destination (O-D) matrix. Its first row contains port identification numbers (every port must be included; the first row cannot have identification numbers for non-port nodes) Its first column contains the identification number of every node in the system (including non-port nodes as well as port nodes). The order in which the identification numbers are entered does not matter. Each row, second through *n*th columns, contains identification numbers of the nodes to which a vessel must next transit in going from its current node (in column one) to its destination (in row one). A zero is entered in the row 1, column 1 position

4.2 TABLE OF NEXT NODES, section 2. The first row and first column of the second section must be identical to those of the first section The body of the table contains identification numbers of sets of parallel links whenever a set exists between a node and a destination. When no set of parallel links exists, enter a value of zero

4.3 FACILITIES ID TABLE contains identification numbers for every link between every two nodes in the system The first row and first column both contain identification numbers for each node (ports and non-port nodes) in the system The order of these identification numbers is unimportant Columns two through *n* of rows two through *n* contain the identification numbers of the link between any two respective nodes A zero is entered in the row 1, column 1 position

4.4 Number of rows in PARALLEL FACILITIES TABLE Optional Omit if datum 1 4 is zero

4.5 PARALLEL FACILITIES TABLE Optional. Omit if datum 1 4 is zero Its table is read by rows (a row is not a card. A row contains all the information for any single alternative) Each row must contain the following four items, in proper order:

(1) identification number of the set of parallel facilities

(2) the number of parallel alternatives within the set, i e , how many choices does a vessel have?

(3) the alternative within each set through which all traffic is to transit during an EDB run The value given here corresponds to the position within a set in which the alternative is listed If the alternative which is listed first in the set is to be used, the value is 1; if the alternative which is listed second in the set is to be used, the value is 2, etc. Enter this even for an EVENT LOG run

(4. The number of nodes included in the alternative. Every set of parallel facilities must begin and end with a node, and must have one node separating each

Datum Description

pair of adjacent links within the alternative. The total number of these nodes (ports included) is given here

The remainder of each row is determined solely by the particular alternative being described. Item number five in each row is the identification number of the alternative's upstream end node. The sixth item is the identification number of the link between that end node and the next node in the alternative. The items continue in this manner until the alternative is completely described in terms of node, link, node, link, node, etc.

4.6 Alternative-wide coefficients for expected transit time functions (Optional Omit if 1.2. = 1 and/or 1.4. = 0.) There are six items for each row in the PARALLEL FACILITIES TABLE:

(1) identification number of the set of parallel facilities
(2) identification number of the alternative's upstream end node from which these coefficients apply (this gives consideration to directional effects)
(3) coefficient to weight the effect upon a vessel's transit time through the alternative of the length of the vessel
(4) coefficient to weight the effect upon a vessel's transit time through the alternative of the previous vessel's expected transit time
(5) coefficient to weight the effect upon a vessel's transit time through the alternative of the total number of vessels in transit through the alternative, regardless of direction
(6) constant for expected alternative transit time function. If none, set to zero.

The six items are repeated, with item number 2 being the downstream end point node for the alternative.

5. Externally-Supplied Fleet Characteristics (Optional: Supply only if datum 1.8 is 1 (one)

When vessel arrivals into the simulated system are to be triggered from an external source, the following nine items are provided on the input device given in 1.6 for each vessel arrival:

Datum Description

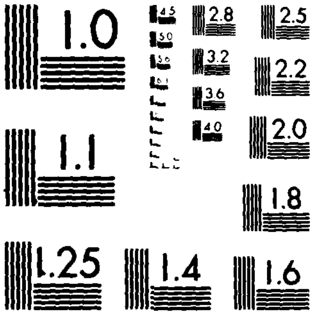
5.1 the event name, PORT EXIT

5.2 time of arrival (must be given as a <u>real</u> number; <u>not integer</u>).

AD-A111 753   PENNSYLVANIA TRANSPORTATION AND TRAFFIC SAFETY CENTER--ETC   F/G 13/10
GREAT LAKES SIMULATION STUDIES. VOLUME I. NETSIM: A GENERAL NET--ETC(U)
NOV 72   DACW23-72-C-0066
UNCLASSIFIED   TTSC-7215   NL

2 of 2
AD
A111753

END
DATE
FILMED
4-82
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Datum	Description
5.3	vessel identification number (1-9999).
5.4	overall length of vessel (in feet).
5.5	commodity code (1-9999).
5.6	identification number of port from which vessel is to initially enter system.
5.7	a code signifying whether the vessel has a pre-determined itinerary (1, if itinerary, 0, if no itinerary).
5.8	number of ports on itinerary. Optional. Omit if Datum 5.7 is 0.
5.9	itinerary. Optional. Vessel identification number and *n* port identification numbers. Omit if 5.7 is 0 (refer to 3.53).
5.10	asterisk (*)

C. Output Data Bases

NETSIM/SHIP produces as output the following two data bases:

(1) the Experience Data Base (EDB)

(2) the Simulation *Event LOG*

The basic difference between the two bases involves the model's assignment decision rule. The EDB is analyzed statistically to produce coefficients that are used in expected transit time functions, during a subsequent simulation run. These functions are used to make the channel choice assignment decision.

1. The Experience Data Base (EDB)

The EDB consists basically of 2 parts: the first gives *prior* conditions existing within an alternative at the point in time when a vessel's assignment decision is to be made, while the second gives *subsequent* arrival and exit times for the vessel's transit through each link within the chosen alternative.

The EDB has four different kinds of records, each of which is distinguished below. The first six items of each of the four records are identical. These six items are as follows:

Item	Format	Description
1	I1	Reference code; 1 indicates that data are prior conditions, 2 indicates that data are subsequent arrival and exit times.
2	I3	Number of nodes remaining in the alternative.
3	I1	Facility type code: 1 = lock, 2 = reach, 3 = lake, 4 = port.
4	I4	Facility identification number.
5	I4	Vessel identification number.
6	I4	Overall length of vessel.

When the reference code is 1, indicating data on prior conditions, there are three possibilities for the remaining items of the record, depending upon the types of facilities within the alternative. These three possibilities are as follows:

For Facility Type Code = 1 (indicating a lock):

Item	Format	Description
7	I4	Number of vessels in near queue.
8	I1	0, if there is no vessel in near throat. 1, if there is a vessel in near throat moving in opposite direction. 2, if there is a vessel in near throat moving in same direction.
9	I1	0, if there is no vessel in chamber. 1, if there is a vessel in chamber moving in opposite direction. 2, if there is a vessel in chamber moving in same direction.
10	I1	0, if there is no vessel in far throat. 1, if there is a vessel in far throat moving in opposite direction. 2, if there is a vessel in far throat moving in same direction.

Item	Format	Description
11	I4	Number of vessels in far queue.

For Facility Type Code = 2 (indicating a reach):

Item	Format	Description
7	I4	Number of vessels in reach that are moving in opposite direction.
8	I4	Number of vessels in reach that are moving in same direction.

For Facility Type Code = 3 (indicating a lake):

Item	Format	Description
7	I4	Number of vessels in transit on lake.

When the reference code is 2, indicating data on subsequent link arrival and exit times, each record has the same data items. The first six items of each record are as discussed above. The remaining items are as follows:

Item	Format	Description
7	I7	Arrival time.
8	D(7,0)	Exit time.

2. The Simulation Event LOG

During a simulation run, a record of every event that occurs is kept in the EVENT LOG. Each record contains the following nine items:

Item	Format	Description
1	I4	Vessel identification number.
2	I4	Vessel length, in feet.
3	I4	Identification number of vessel's current node.
4	I4	Identification number of vessel's next node.

Item	Format	Description
5	I1	Facility-type code 1 - lock 2 - reach 3 - lake 4 - port
6	I4	Facility identification number.
7	I4	Event code.
8	D(7,0)	Current time.
9	I3	Number of vessels in near queue (if relevant).

Item number 7, the event code, is interpreted according to a four-digit code. These codes, along with their interpretations, are given below.

Event Codes

1001 Vessel is at chamber gates ready to enter, but the water level is not quite correct. Hence, the vessel experiences a delay. The eighth parameter (current time) given in the event log is actually the length of this delay.

Decision-triggering event: lock arrival (1000)

Resultant action: enter near queue (1100)

1100 Near queue, near throat, chamber are empty. Water level at the moment is correct, but chamber is recycling to the other side.

1101 Near queue is not empty.

1102 Near queue is empty. Near throat is not empty.

1103 Near queue and near throat are empty. Chamber contains a vessel sailing toward arrival vessel.

1104 Near queue and near throat are empty. Chamber contains a vessel sailing in same direction as arrival vessel. Far queue is not empty.

1105 Near queue and near throat are empty; chamber is empty; water level is not okay; far throat is empty; far queue is not empty.

1106 Near queue and near throat are empty; chamber is empty; water level is not okay; far throat contains a vessel sailing in same direction as arrival vessel; far queue is not empty.

1107 Near queue and near throat are empty; chamber is empty; water level is not okay; far throat contains a vessel sailing toward arrival vessel.

1108 Near queue and near throat are empty; chamber contains a vessel sailing in same direction as arrival vessel; far queue is empty; service look-ahead feature is used; far reach contains a vessel which can sail into chamber before arrival vessel; reach vessel uses MOVING.TTT and arrival vessel uses SHORT.TTT for comparison.

1109 Same as 1108 above except that arrival vessel uses MOVING.TTT.

1110 Same as 1108 above except that reach vessel uses STATIONARY.TTT.

1111 Same as 1108 above except that reach vessel uses STATIONARY.TTT and arrival vessel uses MOVING.TTT.

1112 Near queue and near throat are empty; chamber is empty and is not now scheduled for recycling; water level is not okay; far throat contains a vessel sailing in same direction as arrival vessel; service look-ahead feature is used; chamber has not been free long enough to complete recycling before arrival could be in chamber; far reach contains a vessel which can sail into chamber before arrival; reach vessel uses STATIONARY.TTT and arrival vessel uses SHORT.TTT for comparison.

1113 Same as 1112 above except that arrival uses MOVING.TTT.

1114 Same as 1112 above except that reach vessel uses MOVING.TTT.

1115 Same as 1112 above except that reach vessel uses MOVING.TTT and arrival vessel uses MOVING.TTT.

1116 Near queue and near throat are empty; chamber is empty and is not now scheduled for recycling; water level is not okay; far throat is empty; service look-ahead feature is used; chamber has not been free long enough to have now recycled; chamber has not beeen free long enough to complete recycling before arrival vessel could enter chamber; far reach contains a vessel which can sail into chamber before arrival vessel; reach vessel uses MOVING.TTT and arrival vessel uses SHORT.TTT.

1117 Same as 1116 above except that chamber has been free long enough to complete recycling before arrival could enter chamber, so that arrival vessel uses MOVING.TTT.

1118 Same as 1116 above except that chamber has been free long enough to have now recycled, so that arrival vessel uses MOVING.TTT.

Decision-triggering event: lock arrival (1000)

Resultant action: move to short-entry position (1200)

1201 Near queue and near throat are empty; chamber contains a vessel sailing in same direction as arrival vessel; far queue is empty; service look-ahead feature is not used; chamber is now recycling but is not able to recycle before vessel would enter chamber using MOVING.ITT.

1202 Near queue, near throat, chamber and far queue are empty; service look-ahead feature is not used; chamber is recycling but will not complete recycle before arrival could move into chamber.

1203 Near queue, near throat, chamber and far queue are empty; service look-ahead feature is not used; chamber is not now recycling; chamber is now scheduled to recycle, but recycling would not be completed before arrival vessel could move into chamber.

1204 Near queue, near throat, chamber, and far queue are empty; service and look-ahead feature is not used; chamber is not now cycling nor is it now scheduled for recycling; chamber has not been free long enough to have completed recycling before arrival vessel could move into chamber.

1205 Near queue, near throat, and far queue are empty; chamber contains a vessel moving in same direction as arrival vessel; service look-ahead feature is used; next reach is empty; chamber would not be able to complete process and recycle before vessel could move into chamber.

1206 Near queue, near throat, chamber, and far queue are empty; service look-ahead feature is used; next reach is empty; chamber is now scheduled for recycling, but it will not be completed before arrival vessel could move into chamber.

1207 Near queue, near throat, chamber, and far queue are empty; service look-ahead feature is used; next reach is empty; chamber is not now scheduled for recycle, nor is it now recycling, chamber has not been free long enough to have completed recycling before arrival vessel could move into chamber.

1208 Near queue, near throat, chamber, and far queue are empty, service look-ahead feature is used; next reach is empty; chamber is now recycling, but will not complete recycle before arrival vessel could move into chamber.

1209 Near queue, near throat, and far queue are empty; chamber contains a vessel moving in same direction as arrival vessel; service look-ahead feature is used; next reach contains a vessel sailing toward arrival vessel; vessel from next reach will arrive at clear point after chamber vessel (therefore uses MOVING.TTT); chamber cannot complete processing and recycle before arrival could have moved into chamber (therefore uses SHORT.TTT); vessel from next reach cannot be in chamber before arrival vessel.

1210 Near queue, near throat, and far queue are empty; chamber contains a vessel moving in same direction as arrival vessel; service look-ahead feature is used; next reach contains a vessel sailing toward arrival vessel; reach vessel will arive at clear point before chamber vessel (therefore using STATIONARY.TTT); chamber cannot complete processing and recycle before arrival could have moved into chamber (therefore uses SHORT.TTT); reach vessel cannot be in chamber before arrival vessel.

1211 Near queue, near throat, chamber, and far queue are empty; service look-ahead feature is used; next reach contains a vessel sailing toward arrival; chamber is now recycling, but will not finish before arrival could have moved into chamber.

1212 Same as 1211 except that chamber is not now recycling; chamber has been scheduled for recycling, but will not complete recycle before arrival vessel could have moved into chamber.

1213 Near queue, near throat, chamber, and far queue are empty; service look-ahead feature is used; next reach contains a vessel sailing toward arrival vessel; chamber is not now scheduled for recycling nor is it now recycling; far throat contains a vessel sailing in same direction as arrival vessel; reach vessel will arrive at clear point before throat vessel (therefore reach vessel uses STATIONARY.TTT); chamber has not been free long enough to have recycled before arrival vessel could have moved into chamber (therefore arrival vessel uses SHORT.TTT); reach vessel cannot be in chamber before arrival vessel.

1214 Same as 1213 except that reach vessel will not arrive at clear point before throat vessel (therefore reach vessel uses MOVING.TTT).

1215 Near queue, near throat, chamber, far throat, and far queue are empty; service look-ahead feature is used, next reach contains a vessel sailing toward arrival vessel; chamber is not now recycling nor is it now scheduled to recycle; chamber has not been free long enough to have now recycled; chamber has not been free long enough to have recycled before arrival could be in chamber (so arrival uses SHORT.TTT); reach vessel cannot be in chamber before arrival vessel.

Decision-triggering event: lock arrival (1100)

Resultant action: move directly into chamber (1300)

1301 Near queue and near throat are empty; chamber contains a vessel sailing in same direction as arrival vessel; far queue is empty; service look-ahead feature is not used; chamber is now recycling and will complete cycle before arrival could move into chamber.

1302 Near queue, near throat, chamber, and far queue are empty; far throat is either empty or contains a vessel sailing in same direction as arrival vessel; service look-ahead feature is not used; chamber is now recycling and will complete cycle before arrival vessel could have moved into chamber.

1303 Same as 1302 except that chamber is not now recycling, chamber is now scheduled to recycle and will complete recycling before arrival vessel could have moved into chamber.

1304 Same as 1302 except that chamber is neither now recycling nor scheduled for recycling; chamber has been free long enough to have recycled before arrival vessel could have entered chamber.

1305 Near queue, near throat, and far queue are empty; chamber contains a vessel sailing in same direction as arrival vessel; service look-ahead feature is used; next reach has no vessel sailing toward arrival vessel; chamber is now processing and will be able to complete process and recycle before arrival could move into chamber.

1306 Near queue, near throat, chamber and far queue are empty; look-ahead feature is used; next reach contains no vessel sailing toward arrival vessel; chamber is now scheduled to recycle and will complete recycling before arrival vessel could move into chamber.

1307 Same as 1306 except that chamber is neither now scheduled to recycle nor is now recycling; chamber has been free long enough to have completed recycling before arrival vessel could move into chamber.

1308 Same as 1306 except that chamber is not now scheduled to recycle but is now recycling and will complete recycling before arrival vessel could move into chamber.

1309 Near queue and near throat are empty; chamber contains a vessel sailing in same direction as arrival vessel, service look-ahead feature is used; next reach contains a vessel sailing toward arrival vessel; reach vessel arrives at clear point after chamber vessel (reach vessel uses MOVING.III); chamber is able to complete processing and recycle before arrival vessel could have entered chamber (arrival vessel uses MOVING.III); reach vessel cannot be in chamber before arrival vessel.

1310 Near queue and near throat are empty; chamber contains a vessel sailing in same direction as arrival vessel; service look-ahead feature is used, next reach contains a vessel sailing toward arrival vessel; reach vessel arrives at clear point before chamber vessel (reach vessel, therefore, uses STATIONARY.III); chamber is able to complete processing and recycle before arrival vessel could have entered chamber (arrival vessel, therefore, uses MOVING.III); reach vessel cannot be in chamber before arrival vessel.

1311 Near queue, near throat, chamber and far queue are empty; service look-ahead feature is used; next reach contains a vessel sailing toward arrival; chamber is now recycling and will complete cycle before arrival vessel could have moved into chamber

1312 Near queue, near throat, chamber and far queue are empty; service look-ahead feature is used; next reach contains a vessel sailing toward arrival; chamber is not now recycling, but is now scheduled for recycling and will complete recycle before arrival vessel could move into chamber.

1313 Near queue, near throat, chamber, and far queue are empty; far throat contains a vessel sailing in same direction as arrival vessel; service look-ahead feature is used; next reach contains a vessel sailing toward arrival vessel; chamber is not now recycling, nor is it now scheduled for recycling, reach vessel arrives at clear point before throat vessel (reach vessel, therefore, uses STATIONARY.III); chamber has been free long enough to have recycled before arrival vessel could have entered chamber (arrival vessel, therefore, uses MOVING.III); reach vessel cannot be in chamber before arrival vessel.

1314 Same as 1313 except that reach vessel does not arrive at clear point before throat vessel (reach vessel, therefore, uses STATIONARY.III).

1315 Near queue, near throat, chamber, far throat, and far queue are empty; service look-ahead feature is used; next reach contains a vessel sailing toward arrival vessel; chamber has not been free long enough to have now recycled, but it has been free long enough to have completed recycling before arrival vessel could have entered chamber; reach vessel cannot be in chamber before arrival vessel.

1316 Near queue, near throat, chamber, far throat, and far queue are empty; service look-ahead feature is used; next reach contains a vessel sailing toward arrival vessel; chamber has been free long enough to have now recycled; reach vessel cannot be in chamber before arrival vessel.

1317 Near queue, near throat, and chamber are empty; water level is okay. Service look ahead feature is used.

Decision-triggering event: exit queue (2000)

Resultant action: move into chamber (2100)

2101 Chamber is empty. Water level is okay.

2102 Chamber is empty. Water level is not okay. Chamber is now recycling and will complete cycle before vessel could move into chamber.

2103 Chamber is empty. Water level is not okay. Chamber is not now recycling. Chamber is now scheduled to begin recycling and will complete recycling before vessel could move into chamber.

Decision-triggering event: exit queue (2000)

Resultant action: move into short-entry position (2200)

2201 Chamber is not empty.

2202 Chamber is empty. Water level is not okay. Chamber is now recycling but will not complete cycle before vessel could move into chamber.

2203 Chamber is empty. Water level is not okay. Chamber is not now recycling. Chamber is now scheduled to begin recycling but will not complete recycling before vessel could move into chamber.

Decision-triggering event: move to short entry position (3000)

3101 Vessel arrives at short-entry position.

Decision-triggering event: transit throat (4000)

4101 Vessel enters chamber.

4102 Vessel exits to clearance point.

Decision-triggering event: end cycle (6000)

Resultant action: move to gates-clear point (6100)

6101 Chamber is not empty.

Decision-triggering event: end cycle (6000)

Resultant action: move from short-entry position into chamber (6200)

6201 Chamber recycled empty. Throat to which gates have just opened contains a vessel.

Decision-triggering event: end cycle (6000)

Resultant action: chamber remains open (6300)

6301 Chamber recycled empty. Throat and queue to which gates have just opened are both empty. Since there is no vessel in the immediate vicinity, the vessel characteristics in the event log record are artificially given the values 9999.

Decision-triggering event: end cycle (6000)

Resultant action: chamber remains open (6400)

6401 Chamber recycled empty. Throat to which gates have just opened contains a vessel moving into chamber, i.e., it is not waiting in short-entry position.

Decision-triggering event: pass through gates clear point (7000)

Resultant action: chamber vessel exits to clearance point (7100)

7101 Throat behind chamber vessel is empty. Queue in front is empty. Queue in back is not empty. Service look-ahead feature is used. Next reach contains a vessel sailing toward lock which could move into chamber before queued vessel could.

7102 Throat behind chamber vessel is empty. Queue in front is not empty.

7103 Throat behind chamber vessel is empty. Queue in front and queue in back are both empty. Service look-ahead feature is used. Previous reach and next reach both contain vessels sailing toward lock. Vessel from next reach can move into chamber before vessel from previous reach could.

7104 Throat behind chamber vessel is empty. Queue in front and queue in back are both empty. Service look-ahead feature is used. Previous reach does not contain a vessel sailing toward lock.

7105 Throat behind chamber vessel is empty. Queue in front and queue in back are both empty. Service look-ahead feature is not used.

Decision-triggering event: pass through gates clear point (7000)

Resultant action: chamber begins recycle when gates are cleared.
Chamber vessel exits to clearance point (7200)

7201 Throat behind chamber vessel is not empty.

7203 Throat behind chamber vessel is empty. Queue in front and queue in back are both empty. Service look-ahead feature is used. Previous reach and next reach both contain vessels sailing toward lock. Vessel from next reach cannot move into chamber before vessel from previous reach could.

7204 Throat behind chamber vessel is empty. Queue in front and queue in back are both empty. Service look-ahead feature is used. Previous reach contains a vessel sailing toward lock. Next reach does not contain a vessel sailing toward lock.

Decision-triggering event: pass through gates clear point (7000)

Resultant action: queue vessel exits queue. Chamber begins recycle when gates are cleared. Chamber vessel exits to clearance point (7300)

7301 Throat behind chamber vessel is empty. Queue in front is empty. Queue in back is not empty. Service look-ahead feature is used. Next reach does not contain a vessel sailing toward lock.

7302 Throat behind chamber vessel is empty. Queue in front is empty. Queue in back is not empty. Service look-ahead feature is not used

7303 Throat behind chamber vessel is empty. Queue in front is empty. Queue in back is not empty. Service look-ahead feature is used. Next reach contains a vessel sailing toward lock which cannot move into chamber before queued vessel could.

Sail Across Lake (7100)

7110 Vessel enters lake

Exit Lake (7200)

7210 Vessel exits lake

Sail Through Reach (8100)

8101 Vessel enters reach

Exit Reach (8200)

8201 Vessel exits reach.

Enter Port (9100)

9101 Vessel enters port

Exit Port (9200)

9211 Vessel selects next port from itinerary.

9221 Vessel enters berth.

9231 Vessel selects next port from scheduler.

3. Error Messages

The error messages produced by NETSIM/SHIP are interpreted through four-digit codes. Each error code, along with its cause, is given below.

Error Number Cause

In INITIALIZATION ROUTINE:

3001 A FOR EACH PORT statement did not locate a port whose identification number was equal to that of the vessel's current node.

In MOVEMENT CONTROLLER routine:

4101 A FOR EACH PORT statement did not locate a port whose identification number was equal to that of the vessel's destination.

4104 In searching the first column of the PARALLEL.FACILITIES.TABLE, no identification number was found which matched that given by the TABLE.OF.NEXT.NODES.

4105 The variable NEXT.FACILITY obtained from the FACILITIES.TABLE does not match any reach, lock, or lake.

4106 No identification number was found in the 5th column or Pth column of the Lth row of the PARALLEL.FACILITIES.TABLE which was the same as that for the vessel's current node

In ALTERNATIVE.SELECTOR routine:

4201 During an EDB run, one alternative within a set was found to be impassable because of a lock's maximum vessel size.

4202 During a LOG run, every alternative within a set was found to be impassable because of locks' maximum vessel sizes.

4203 In searching through the array of expected transit times, no match was found for the least expected transit time.

4204 The facility identification number given in the Dth row, Fth column, of the PARALLEL.FACILITIES.TABLE was not found to match that of any lock, reach, or lake in the system.

In the ROUTINE TO SEARCH.TABLE.OF.NEXT.NODES:

5101 In searching the first row of the TABLE.OF.NEXT.NODES, no identification number was found to match the argument D.

5102 In searching the first column of the TABLE.OF.NEXT.NODES, no identification number was found to match the argument C.

In the ROUTINE TO REFERENCE.FACILITIES.TABLE:

5201 In searching the first row of the FACILITIES.ID.TABLE, no identification number was found to match the argument B.

5202 In searching the first column of the FACILITIES.ID TABLE, no identification number was found to match the argument A.

In the ROUTINE FOR STOCHASTIC.TIME.CALCULATIONS:

5301 In searching the first column of the array of lake ETT coefficients, no identification number was found to match that of the current lake.

5302 In searching the array of reach ETT coefficients, no identification number was found to match that of the current reach.

5303 In searching the array of lock ETT coefficients, no identification number was found to match that of the current lock.

5304 In searching the array of parallel set ETT coefficients, no identification number was found to match that of the current parallel set.

5401 In searching the array of arguments for a stationary chamber entry, no identification number was found to match that of the current lock.

5402 In searching the array of arguments for a moving chamber entry, no identification number was found to match that of the current lock.

5403 In searching the array of arguments for a short chamber entry, no identification number was found to match that of the current lock.

5404 In searching the array of arguments for a lock recycle, no identification number was found to match that of the current lock.

5405 In searching the array of arguments for a lock process, no identification number was found to match that of the current lock.

5406 In searching the array of arguments for port time, no identification number was found to match that of the current port.

5407 In searching the array of arguments for lake transit times, no identification number was found to match that of the current lake.

5408 In searching the array of arguments for chamber exit times, no identification number was found to match that of the current lock.

5409 In searching the array of arguments for lock exit times, no identification number was found to match that of the current lock.

5410 In searching the array of arguments for reach transit times, no identification number was found to match that of the current reach.

5411 In searching the array of arguments for arrival-to-short-entry-position times, no identification number was found to match that of the current lock.

5412 In searching the array of arguments for queue-to-short-entry-position times, no identification number was found to match that of the current lock.

In the ROUTINE TO QUERY.INTRALAKE.DISTANCES.TABLE

5701 In searching the INTRALAKE.DISTANCES.TABLE, no identification number was found to match that given by the variable NXT.NODE.

5702 In searching the INTRALAKE.DISTANCES.TABLE, no identification number was found to match that of the current vessel's current node.

5703 In searching the INTRALAKE.DISTANCES.TABLE, no identification number was found to match that of the current lake.

In the ROUTINE TO ARRIVE.AT.LOCK:

6101 A FOR EACH END.CYCLE statement failed to find an END.CYCLE event notice for the current lock (the current lock is not now processing, although there is a vessel now in the chamber).

6102 A FOR EACH END.CYCLE statement failed to find an END.CYCLE event notice for the current lock (the current lock is not now processing, although there is a vessel now in the chamber).

6103 A FOR EACH END.CYCLE statement failed to find an END.CYCLE event notice for the current lock (the current lock is not now processing, although there is a vessel now in the chamber).

In the EVENT TO EXIT.QUEUE:

6201 A FOR EACH BEGIN.CYCLE statement failed to find a BEGIN.CYCLE event notice for the current lock (the chamber is empty, open to the side opposite the current vessel, and is neither now recycling, nor now scheduled to begin recycling).

6202 A FOR EACH LOCK statement failed to find a lock whose identification number was equal to that given by the variable LOK.ID.

In the EVENT TO MOVE.TO.SHORT.ENTRY.POSITION:

6301 A FOR EACH LOCK statement failed to find a lock whose identification number was equal to that given by the variable ID.LOK.

In the EVENT.TO.TRANSIT.THROAT.

6401 A FOR EACH LOCK statement failed to find a lock whose identification number was equal to that given by the variable ID.LOK.

In the EVENT.TO.BEGIN.CYCLE.

6501 A FOR EACH LOCK statement failed to find a lock whose identification number was equal to that given by the variable ID.LOK

In the EVENT.TO.END.CYCLE:

6601 A FOR EACH LOCK statement failed to find a lock whose identification number was equal to that given by the variable LCK.ID.

In the EVENT.TO.PASS.THROUGH.GATES.CLEAR.POINT:

6801 A FOR EACH LOCK statement failed to find a lock whose identification number was equal to that given by the variable LK.ID.

In the EVENT.TO.EXT.LAKE:

7201 A FOR EACH VESSEL statement failed to find a vessel in the set ON.LAKE.TRAFFIC whose identification number was equal to that given by the variable LAK.ID.

In the EVENT.TO.LEAVE.REACH:

8201 A FOR EACH REACH statement failed to find a reach whose identification number was equal to that given by the variable REA.ID.

In the EVENT.FOR.PORT.EXIT:

9201 A FOR EACH VESSEL statement failed to find a vessel, in the set DOCK, whose identification number was equal to that given by the variable ID.PRT.

9203 A FOR EACH BERTH statement failed to find a berth, in the set SET.CONTAINING.BERTHS, whose identification number was equal to that given by the variable ID.PRT.

ILMED

8